T.C.

MİLLÎ EĞİTİM BAKANLIĞI

ENDÜSTRİYEL OTOMASYON TEKNOLOJİLERİ

İNTERNET ÜZERİNDEN ÜRÜN TAKİBİ

Ankara, 2014

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- PARA İLE SATILMAZ.

İÇİNDEKİLER

AÇIKLAMALAR iv
1. PROGRAMLAMA İLE SİSTEMDEN VERİ ALMA 3
1.1. Sistemin Hazırlanması 3
1.1.1. MSP430G2231 Mikro Denetleyici 8
1.1.2. P1DIR Kaydedicisi 8
1.1.3. P1REN Kaydedicisi 8
1.1.4. P1OUT Kaydedicisi 8
1.1.5. P1IN Kaydedicisi 9
1.1.6. P1 Portu Kesmeleri 9
1.1.7. P1IFG (P1 Interrupt Flag) Kaydedicisi 9
1.1.8. P1IE (P1 Interrupt Enable) Kaydedicisi 9
1.1.9. P1IES (P1 Interrupt Edge Select) Kenar Seçme Kaydedicisi 10
1.1.10. MSP430 Clock Modülü (İşaret Kaynakları) 10
1.1.12. MSP430 ADC10 Modülü 12
1.1.13. MSP430 Low Power Özelliği 13
1.2. Sistem Kontrolörünün programlanması 15
1.3. Programlama Kontrolör Bağlantısı 26
1.4. Kontrolörden Alınan Verilerin Text Tipi Dosyaya Kaydedilmesi 31
UYGULAMA FAALİYETİ 36
UYGULAMA FAALİYETİ 38
UYGULAMA FAALİYETİ 39
ÖLÇME VE DEĞERLENDİRME 42
ÖĞRENME FAALİYETİ–2 44
2. BETİK DİLİ İLE VERİLERİN WEB‘E AKTARILMASI 44
2.1. Betik Dili Text Tipi Dosyadan Veri Okuma 44
ÖĞRENME FAALİYETİ–2 44
2.2. Betik Dili İle Verilerin Web e Aktarılması 50
UYGULAMA FAALİYETİ 52
UYGULAMA FAALİYETİ 53
UYGULAMA FAALİYETİ 54
UYGULAMA FAALİYETİ 56
ÖLÇME VE DEĞERLENDİRME 61
MODÜL DEĞERLENDİRME 62
CEVAP ANAHTARLARI 63
KAYNAKÇA 65

AÇIKLAMALAR

ALAN	Endüstriyel Otomasyon Teknolojileri
DAL/MESLEK	Endüstriyel Kontrol
MODÜLÜN ADI	İnternet Üzerinden Ürün Takibi
MODÜLÜN TANIMI	Görsel programlama kullanarak, kontrol kartları ile ürünlerden topladıkları verileri betik dili ile internet üzerinden takip işlemlerinin yer aldığı öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	İnternet Programcılığı Uygulamaları modülünü almış olmak.
YETERLİK	Kontrol kartlarını internet programcılığında kullanmak.
MODÜLÜN AMACI	**Genel Amaç** Öğrenci, bu modül ile gerekli kontrol kartları ve ortam sağlandığında, görsel programlama ile internet programcılığı konularını aynı anda kullanabileceksiniz. **Amaçlar** **1.** Görsel programlama ile kontrol kartı arasında haberleşmeyi gerçekleştirebileceksiniz. **2.** Görsel programlama ile kontrol kartında elde ettiği bilgileri web sunucusuna kaydetmeyi yapabileceksiniz. **3.** Betik dili ile web sunucusunda olan bir dosyayı, web ortamında gösterilmesini programlayabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	**Ortam** Programlama laboratuvarı, ev, bilgi teknolojileri ortamı vb. **Donanım** USB veya seri port destekli kontrol kartları, güç kaynağı, avometre, çalışma masası, bilgisayar, web-server.
ÖLÇME VE DEĞERLENDİRME	Modülün içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme soruları kendinize ilişkin gözlem ve değerlendirmeleriniz yolu ile kazandığınız bilgi ve becerileri ölçerek kendinizi değerlendireceksiniz. Öğretmen, modül sonunda size ölçme teknikleri uygulayarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Bilgisayar, elektrik ve elektronik artık içiçe geçmiş mesleklerdir. Fabrikaların, elektrik-elektronik kısımlarının revize işlemlerinde, en çok kullanılan enstrüman hiç kuşkusuz kontrol kartlarıdır. Kontrol kartları ile daha ergonomik, istatistiki verilerin veri tabanına aktarımı konuları globalleşen dünyamızda, üretim ve yönetim sisteminin vazgeçilmez unsurlarıdır.

Elinizdeki modül, ülkemizde de bulunabilen temel elektriksel giriş-çıkış işlemlerini bilgisayar destekli yapabilen, kontrol kartlarını daha yakından tanıyabilecek, belki bazı kontrol kartlarına sahip olarak evinizdeki elektrikli ev aletlerinin kontrolünü bilgisayar ve hatta internet üzerinden yapabileceksiniz.

Bu modülde, kontrol kartlarının bilgisayarınıza nasıl bağlanılacağını, görsel programlama ile bu karta nasıl erişebileceğinizi, erişim sonuçlarını veri tabanında toplayıp sonuç bilgilerini, betik dillerini kullanarak web sunucusu üzerinden tüm dünyada görülebilecek web sayfaları üretimini yakından kavrayacaksınız.

ÖĞRENME FAALİYETİ–1

AMAÇ

Görsel programlama kullanarak verileri sistemden okuyabileceksiniz.

ARAŞTIRMA

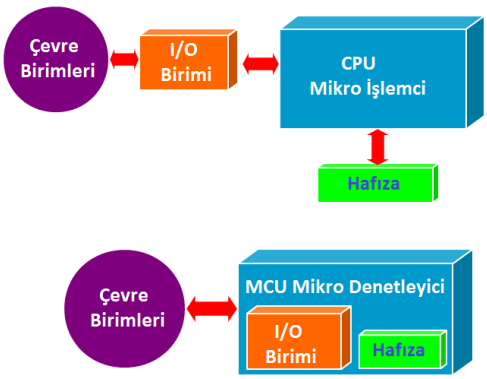
USB veya seri port ile çalışabilen, elektriksel giriş-çıkış (aç-kapa), analog bilgi toplama işlemlerini yapabilecek kontrol kartları hakkında bir ön araştırma yapınız. İşletmelerde bulunabilen bazı cihazların (cnc makinaları, üretim bantları, sıcaklık ve basınç kontrol üniteleri gibi) çalışmasına destek veren kontrol kartları hakkında araştırma yapınız.

1. PROGRAMLAMA İLE SİSTEMDEN VERİ ALMA

Mikro denetleyicileri programlayarak, bir sistemi yönetmek ve bilgi toplamak otomasyon sistemlerinin vazgeçilmez unsurlarıdır. Görsel programlama ile mikro denetleyiciler arasında insan ara yüz programları yazmak sureti ile toplanan bilgileri analiz etmek istatistik çıkarmak temel hedeflerdendir.

1.1. Sistemin Hazırlanması

Mikro işlemci, bellekteki program yardımıyla işlemlerin yapılmasını gerçekleştiren ve aritmetik-mantık, kaydediciler ve bellek birimlerinden oluşan entegre devrelere denir. Mikro denetleyiciler ise, genellikle endüstriye yönelik kontrol ve otomasyon işlemlerini gerçekleştirmek için geliştirilen özel amaçlı mikro işlemcilerdir.

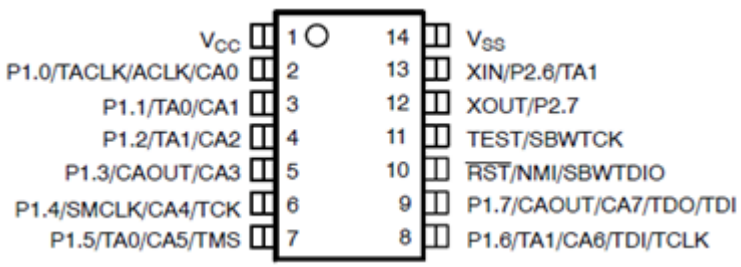


Şekil 1.1 : Mikro işlemci ile mikro denetleyicinin arasındaki fark

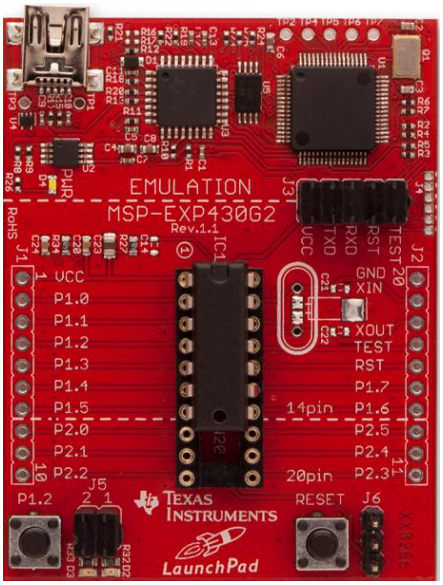
Günümüzde mikro denetleyiciler, basit ve ucuz üretim maliyetlerinden dolayı bir çok firma tarafından üretilmektedir. ARM, MSP430, 8051, AVR, PIC gibi. Mikro denetleyiciler farklı firmaların ürettikleri 8, 16 veya 32 bitlik mikro denetleyiciler vardır. Mikro denetleyicilerin çalışması; devreye güç verdiğinizde mikro denetleyiciye bağlı olan quartz kristal veya dâlilî osilatör çalışmaya başlar ve mikro denetleyicinin işlemleri yapabilmesi için gerekli olan osilasyonu üretir. Bu osilasyonun frekansı sistemin çalışma hızını belirler. Program sayacı başlangıç konumundadır ve bu andan itibaren program belleğindeki komutlar icra edilmeye başlar. Program belleğine yazılan kodlar, haricî kontrol sinyallerine göre sistemi kontrol eder.

Uygulamalarımızda kullanacağımız mikro denetleyicimiz ise **MSP430 LaunchPad** geliştirme aracıdır. Kısaca **Kontrolör** diyeceğimiz bu geliştirme kitimiz, Texas Instruments firmasına aittir. Ülkemizde bulunmaktadır. Zengin uygulama içeriğine sahip olan bu kartın bazı özellikleri şunlardır.

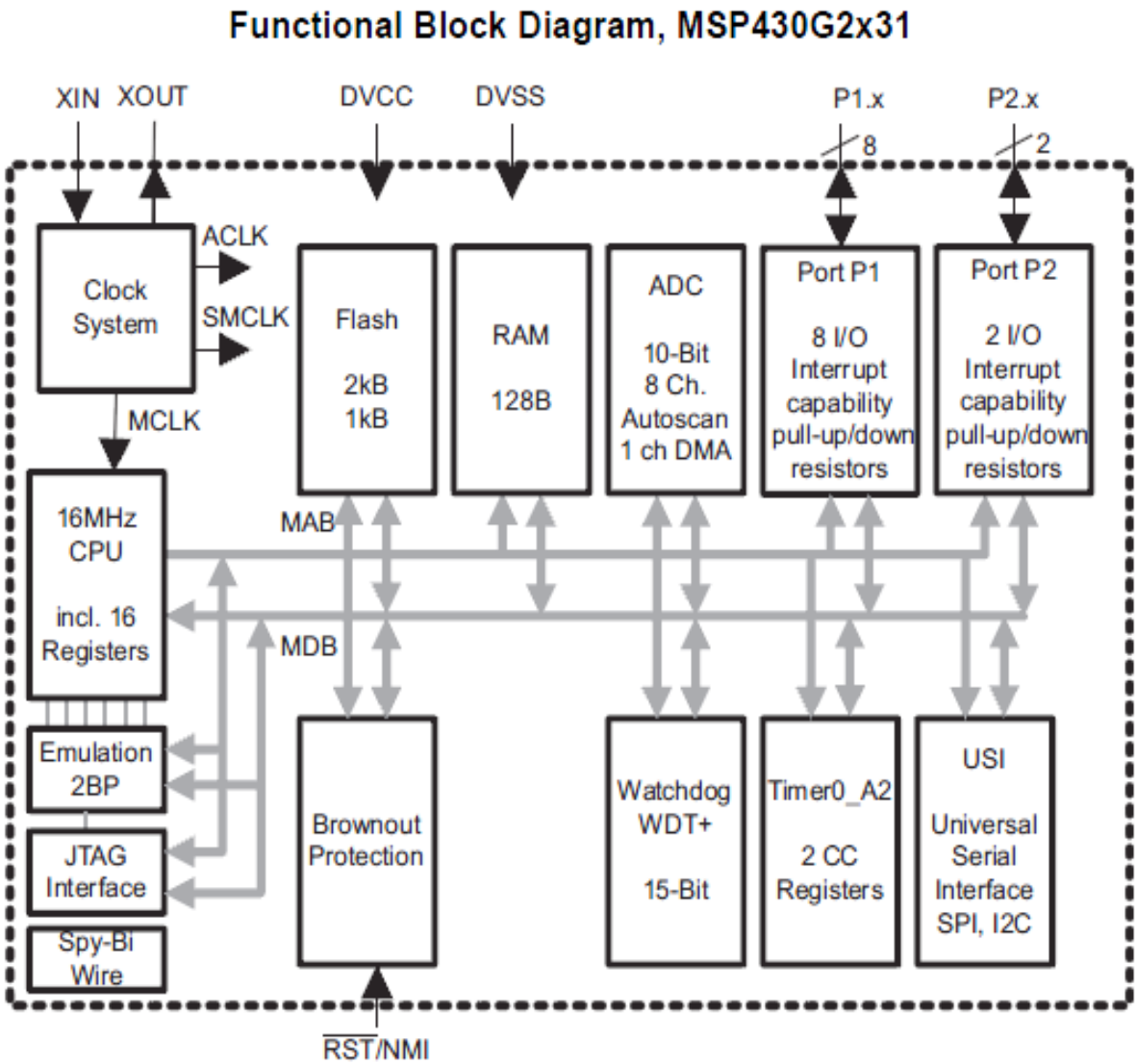
- MSP430G2231 : Düşük güç (Low Power) 16 bitlik işlemci,
- 10 GPIO, çip üzerinde hazır sıcaklık sensörü,
- 8 kanal 10 bit ADC. 2KB flash, 128 byte RAM,
- Üzerinde hazır debugger ve programlayıcı,
- USB üzerinden, başka donanıma ihtiyaç duymadan direk bağlanabilir.
- Kit üzerinden çip değişikliği yapılabilir. Düşük fiyat.



Şekil 1 2 : MSP430 Launch Pad in (Geliştirme Aracı) çip yapısı (Datasheet Sayfa 4)



Resim 1 1 : MSP430 Launch Pad in (Geliştirme Aracı) resmi



Şekil 1.3 : MSP430 Launch Pad geliştirme aracının blok şeması (Datasheet sayfa 5)

MSP430G2231 Pin Fonksiyonları

TERMINAL				DESCRIPTION
NAME	PW or N NO.	RSA NO.	I/O	
P1.0/TACLK/ACLK/CA0	2	1	I/O	General-purpose digital I/O pin Timer_A, clock signal TACLK input ACLK signal ouput Comparator_A+, CA0 input
P1.1/TA0/CA1	3	2	I/O	General-purpose digital I/O pin Timer_A, capture: CCI0A input, compare: Out0 output Comparator_A+, CA1 input
P1.2/TA1/CA2	4	3	I/O	General-purpose digital I/O pin Timer_A, capture: CCI1A input, compare: Out1 output Comparator_A+, CA2 input
P1.3/CAOUT/CA3	5	4	I/O	General-purpose digital I/O pin Comparator_A+, output / CA3 input
P1.4/SMCLK/C4/TCK	6	5	I/O	General-purpose digital I/O pin SMCLK signal output Comparator_A+, CA4 input JTAG test clock, input terminal for device programming and test
P1.5/TA0/CA5/TMS	7	6	I/O	General-purpose digital I/O pin Timer_A, compare: Out0 output Comparator_A+, CA5 input JTAG test mode select, input terminal for device programming and test
P1.6/TA1/CA6/TDI/TCLK	8	7	I/O	General-purpose digital I/O pin Timer_A, compare: Out1 output Comparator_A+, CA6 input JTAG test data input or test clock input during programming and test
P1.7/CAOUT/CA7/TDO/TDI†	9	8	I/O	General-purpose digital I/O pin Comparator_A+, output / CA7 input JTAG test data output terminal or test data input during programming and test
XIN/P2.6/TA1	13	12	I/O	Input terminal of crystal oscillator General-purpose digital I/O pin Timer_A, compare: Out1 output
XOUT/P2.7	12	11	I/O	Output terminal of crystal oscillator General-purpose digital I/O pin
$\overline{\text{RST}}$/NMI/SBWTDIO	10	9	I	Reset or nonmaskable interrupt input Spy-Bi-Wire test data input/output during programming and test
TEST/SBWTCK	11	10	I	Selects test mode for JTAG pins on Port1. The device protection fuse is connected to TEST. Spy-Bi-Wire test clock input during programming and test
V_{CC}	1	16		Supply voltage
V_{SS}	14	14		Ground reference
NC	NA	13, 15		Not connected
QFN Pad	NA	Package Pad	NA	QFN package pad connection to V_{SS} recommended.

† TDO or TDI is selected via JTAG instruction.

NOTE: If XOUT/P2.7 is used as an input, excess current will flow until P2SEL.7 is cleared. This is due to the oscillator output driver connection to this pad after reset.

Şekil 1.4 : MSP430G2231 Mikro Denetleyicisine ait PIN fonksiyonlarını gösterir şema (Datasheet Sayfa 7)

1.1.1. MSP430G2231 Mikro Denetleyici

P1 ve P2 olmak üzere iki adet giriş / çıkış portu vardır. P1 portunda 8 I/O ve P2 portunda 2 I/O pinleri vardır. PIC lerdeki A portu veya B portu gibi. Portların tamamında pull-up veya pull-down dirençleri bulunmaktadır. Herhangi bir portun herhangi bir pininde seviye değişikliği olduğunda (+3V > 0 veya 0 > +3V) program kesme vektörüne yönlendirilebilir.

1.1.2. P1DIR Kaydedicisi

PIC'deki TRIS kaydedicisine benzer. Portun ilgili biti lojik 1 yapıldığında çıkış, Portun ilgili biti lojik 0 yapıldığında ise giriş olarak ayarlanır. Ön tanımlı olarak P1DIR kaydedicisinin değeri 0x00 dır. Başlangıç değeri 0x00 olması ve tüm pinlerin giriş yapılması aslında bir düşük-güç (low power) özelliğidir.

```
P1DIR = 0xF0;
```

İlk dört bit giriş, son dört bit çıkış demektir (**Direction** - 11110000).

1.1.3. P1REN Kaydedicisi

Pull-up = 3V Pull-down = 0V. P1 deki pull-up veya pull-down dirençlerini kontrol etmek için kullanılır. P1REN kaydedicisinin ilgili biti 1 yapıldığında dirençler aktif, 0 yapıldığında ise dirençler pasiftir. Bu kaydedici sadece dirençleri aktif ya da pasif yapar, pull-up veya pull-down olma durumu P1OUT saklayıcısı tarafından kontrol edilir.

```
P1REN=0xF0;
```

İlk dört pindeki dirençler aktif (**Resistor Enable - 11110000**).

1.1.4. P1OUT Kaydedicisi

P1 portunda çıkış olarak ayarlanılan bitleri kontrol eder. P1OUT kaydedicisinin ilgili biti 1 yapıldığında çıkış lojik 1 (~3.3v) P1OUT kaydedicisinin ilgili biti 0 yapıldığında ise çıkış lojik 0 (0v) olur.

```
P1OUT = 0x01;
```

Eğer pin giriş olarak ayarlanmışsa (P1DIR=0x00) P1.0 pini pull-up yapılacak demektir.

1.1.5. P1IN Kaydedicisi

P1 portunun giriş olarak ayalanmış bitlerinin durumunu kontrol etmek için kullanılır. Bu kaydediciden yanlızca okuma işlemi yapılır, yazma yapılamaz (Read only). Bit = 0 ise portun ilgili pini lojik 0 durumunda, Bit = 1 ise portun ilgili pini lojik 1 durumundadır.

```
if (P1IN = 0x01) P1OUT = 0x02;
// P1.0 = 1 ise P1.1 i lojik 1 yap
```

1.1.6. P1 Portu Kesmeleri

MSP430 LaunchPad üzerinde bulunan MSP430G2231 mikro denetleyicisinde toplam 10 adet giriş-çıkış bulunmaktadır. 10 giriş-çıkış da haricî kesmeye gidebilmektedir. Bütün giriş-çıkışlardaki kesme vektör adresi ortak kullanılmaktadır.

1.1.7. P1IFG (P1 Interrupt Flag) Kaydedicisi

P1 portunun ilgili pininde kesme oluşup oluşmadığı, bu saklayıcının bitleri kontrol edilerek öğrenilir. Eğer Bit = 1 ise kesme oluşmuş, Eğer Bit = 0 ise kesme oluşmamış anlamına gelmektedir.

```
if (P1IFG_bit.P1IFG_7 == 1) P1OUT_bit.P1OUT_0 = 1;
//Eğer P1.7 de bir kesme oluşmuş ise P1.0 = 1 yap.
```

1.1.8. P1IE (P1 Interrupt Enable) Kaydedicisi

Bu kaydedici ise P1 portunun ilgili bitinin kesmelere açık olup, olmayacağı durumunu belirlemek için kullanılır. Eğer Bit = 1 ise ilgili pinde kesme açık, eğer Bit = 0 ise ilgili pinde kesme kapalı durumdadır.

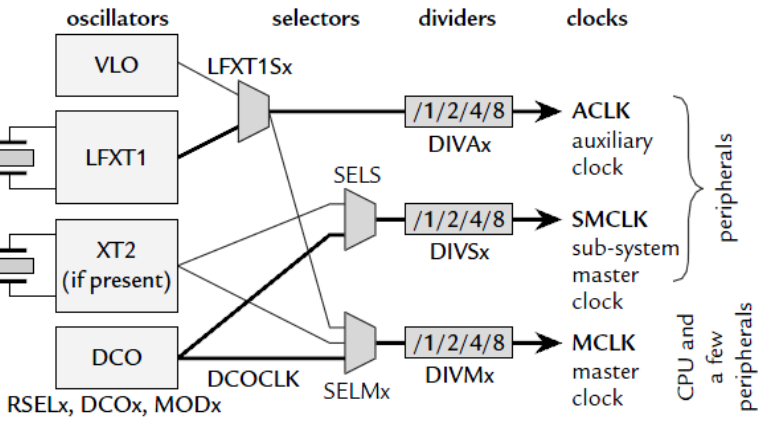
```
P1IE_bit.P1IE_7 = 1;
//P1.7 için harici kesme açık
```

1.1.9. P1IES (P1 Interrupt Edge Select) Kenar Seçme Kaydedicisi

P1 portunun ilgili bitinin kesmeye düşen kenar veya yükselen kenarda gideceğini belirleyen kaydedicidir. Eğer Bit = 1 ise düşen kenar, eğer Bit = 0 ise yükselen kenar.

```
P1IES_bit.P1IES_7 = 1;
//P1.7 için kesmeler düşen kenarda olacak.
```

1.1.10. MSP430 Clock Modülü (İşaret Kaynakları)



Şekil 1.5 : MSP430G2231 mikro denetleyicisinin CLOCK (Saat darbesi) modül yapısı

MSP430 içerisinde 2 si dahilî 1 i haricî olmak üzere toplam 3 farklı clock girişi bulunmaktadır.

- DCOCLK (Digitally Controlled Oscillator) digital kontrollü dâlilî osilatör (yazılım destekli).
- VLOCLK (Very Low Power Clock) çok düşük güç tüketimli dâlilî osilatör.
- LFXT1CLK (Low Frequency / High Frequency External Clock) düşük ve yüksek frekanslı haricî osilatör. MSP430 Launch Pad ile gelen haricî 32,768 KHz lik kristal destekli.
- XT2CLK (High Frequency Clock) eğer sunulmuş ise. MSP430G2231 içerisinde dâhil değildir.

İşaret kaynaklarına kısaca verilen isimler dâhilî olanlar VLO, DCO isimlerini alır. Haricî olan ise LFXT1'dir. Çevre birimlerine bağlanacak çıkışların isimleri ise;

- **ACLK (Auxiliary Clock)** yardımcı işaret kaynağı : Sadece VLO dahilî veya LFXT1 haricî işaret kaynaklarını kullanır,1/2/4/8 e bölünebilir.

- **SMCLK (Sub-System Master Clock)** alt sistem ana işaret kaynağı : Sadece VLO dahilî ile LFXT1 haricî işaret kaynaklarını kullanır ve 1/2/4/8 e bölünebilir.
- **MCLK (Master Clock) Ana İşaret Kaynağı :** 2 dâhilî ve 1 haricî işaret kaynaklarını kullanır ve 1/2/4/8 e bölünebilir.

Şimdi işaret kaynaklarına daha yakından inceleyelim.

1.1.10.1. DCO (Digitally Controlled Oscillator)

Digital kontrollü osilatör: %1 veya %2 hata payı ile tüm MSP430 içerisinde dâhilî olarak bulunan RC osilatördür. IAR yazılım geliştirme programında doğrudan dâhilî frekans elde etmek için

```
BCSCTL1 = CALBC1_1MHZ;  // DCO Birimi 1 MHz Kalibrasyonu set edilir
DCOCTL  = CALDCO_1MHZ;  // DCO Birimi adimi 1 MHz + Kalibrasyon
                        // 1MHZ, 8MHZ, 12MHZ, ve 16MHZ seçenekleri vardır
```

1.1.10.2. VLOCLK (Very Low Power Clock)

Çok düşük güç tüketimli dâhilî osilatör. Hız gerektirmeyen uygulamalar için düşük güçlü uygulamalarda kullanılabilecek bir işaret kaynağıdır. Yazılımla haricî veya dâhilî olarak ayarlanabilir. MSP430 LaunchPad üzerindeki P2.6 ve P2.7 pinleri haricî kristal için giriş pinleridir.

```
BCSCTL3 = LFXT1S_2;     // ACLK = VLO
P2SEL = 0x00;           // P2.6 - P2.7 Harici Kristal
P2DIR = 0x00;           // P2 Portu Giriş
```

1.1.10.3. LFXT1CLK (Low Frequency / High Frequency External Clock)

Düşük ve yüksek frekanslı haricî osilatör : Haricî kristal kaynağıdır. XTS = 0 ve XTS = 1 olmak üzere iki farklı çalışma modu vardır. XTS = 1 Yüksek frekans modudur ve MSP430 LaunchPad'de desteklenmemektedir. XTS = 0 da ise maksimum 32.768 khz lik kristal desteklenmektedir. XCAP bitleri kullanılarak 1pF, 6pF, 10pF ve 12,5pF dâlilî kapasitörlerden biri seçilebilir.

1.1.11. MSP430 Timer_A Modülü

MSP430 Launch Pad in **Timer_A** isimli modülü vardır. Bu modülde;

- 16 Bit timer,
- 2 Capture / Compare saklayıcısı,
- PWM generator (üreteci),
- Farklı işaret kaynakları seçebilme,

- Kesme vektörleri bulunur.

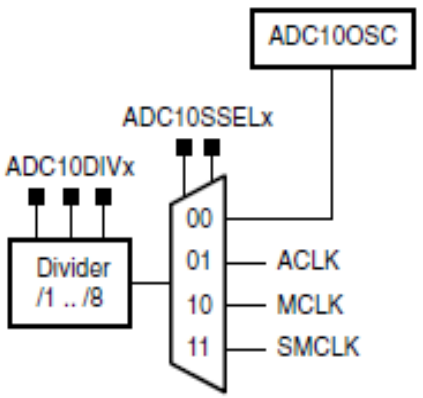
1.1.12. MSP430 ADC10 Modülü

MSP430 Launch Pad in **ADC10** (Analog Digital Converter) modülünde ise;

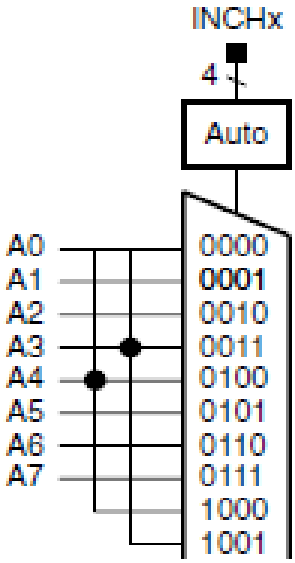
- 10 Bit Analog / Digital dönüşüm,
- 200 ksps,
- 1,5V ve 2,5V dâlilî referans gerilimi üretebilme,
- İşaret kaynağı olarak MCLK, ACLK SMCLK veya ADC10 modülünün kendi işaret kaynağı olan ADC10OSC yı seçebilme,
- Dâlilî sıcaklık sensörü (sonuçlar ADC10MEM saklayıcısında tutuluyor),
- DTC (Data Transfer Controller) veri iletim kontrolörü.

ADC10 modülünün işaret kaynak seçimi için, ADC10SSEL bitlerinden işaret kaynağı seçilmektedir. Frekans bölücüler ise ADC10DIV bitlerinden seçilmektedir. Frekans bölücü; /2, /3, /4, /5, /6, /7, /8 olarak çalışabilmektedir. ADC10OSC = ~1,5 – 6.3 Mhz hıza sahiptir. Analog giriş seçimi INCH bitlerinden yapılmakadır. Giriş seçildikten sonra ise ADC10AE saklayıcısından seçilen giriş için ilgili bit set edilmelidir. Örnek;

```
ADC10CTL1 = INCH_0;       // Kanal 0 Seçildi. Toplam 16 Kanal var.
ADC10AE0   = 1;           // ADC10 Analog 0 Aktif edilir. Pasif için 0
ADC10AE1   = 1;           // ADC10 Analog 1 Aktif edilir. Pasif için 0
```



Şekil 1.6: MSP430 Launch Pad in ADC10 modülüne ait işaret kaynak seçme şeması



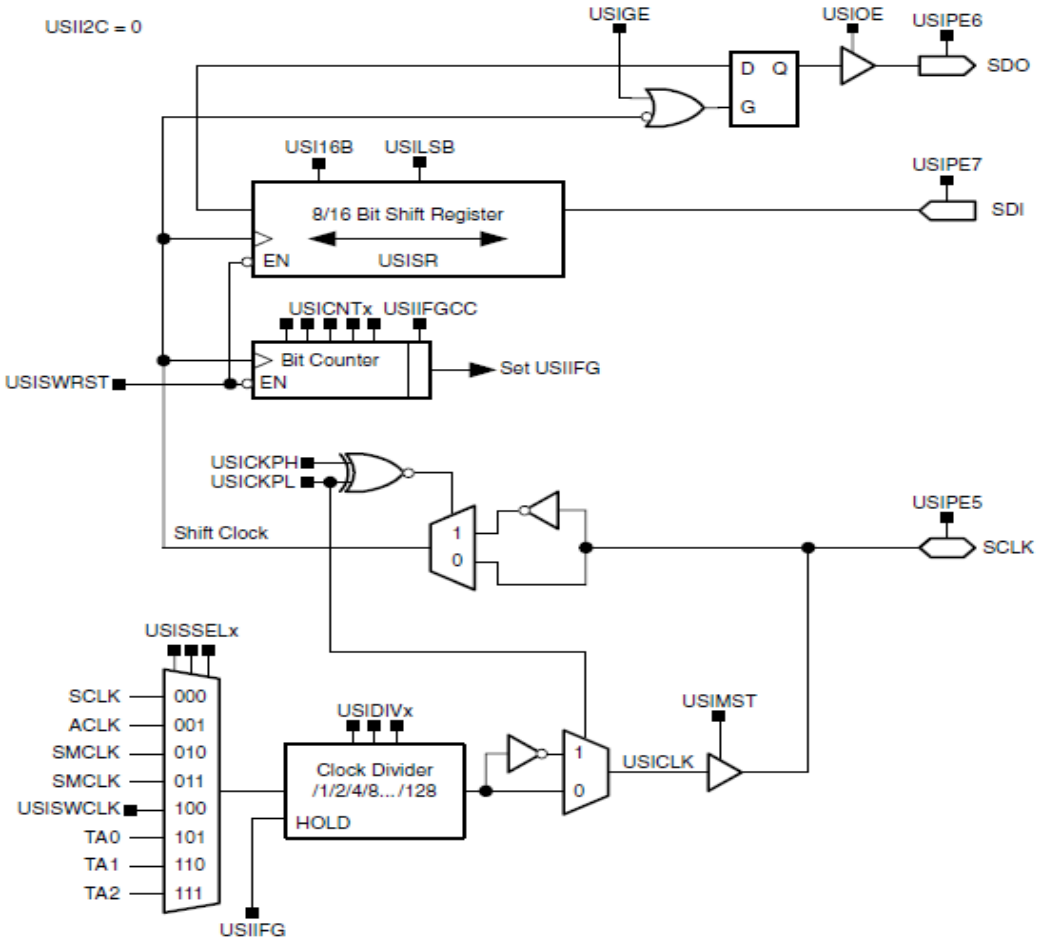
Şekil 1.7 : MSP430 Launch Pad in ADC10 modülüne ait Analog Giriş seçme şeması

1.1.13. MSP430 Low Power Özelliği

ADC10 düşük güç özellikleri ise; Referans gerilimi üreten birim tamamen low power uygulamalar için tasarlanmıştır. 30 US saniyeden az bir zamanda referans gerilimi kullanıma hazır hâle gelmektedir (datasheet). Modül, kullanılmadığı zaman kendini kapatma (auto power down) özelliğine sahiptir.

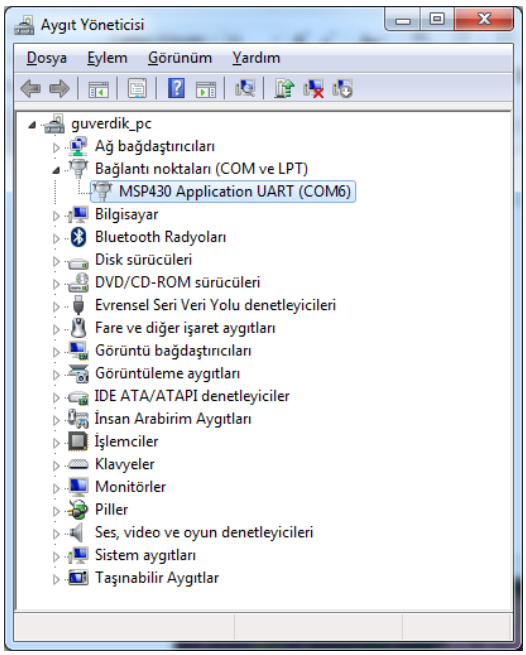
MSP430 Launch Pad in **SPI** (Serial to Parallel Interface) modülünde ise;

- SPI Modunda USI Shift Registeri (USISR),
- 8 ya da 16 bitlik erişim,
- Slave Modunda LPM4 konumunda bekleyebilme (~0.1 uA),
- İşaret kaynağı olarak SMCLK ve ACLK kullanabilme,
- USIDIV_7 ile giriş frekansını 128'e bölebilme,
- USI kesmeleri için atanmış bir adet kesme vektörü,
- I2C işlemlerini uygulayabilme.



Şekil 1.8 : MSP430 Launch Pad in Seri paralel arabirimine ait şema

MSP430 LaunchPad geliştirme aracını USB portuna taktığımız zaman Windows Aygıt Yöneticisinde otomatik olarak tanıtılmış olarak görülür. Geçerli olan port numarasını açılır, seçeneklerden incelenir. Her bilgisayarda farklı gelebilen COM6 gibi port numarası görsel programlarda kullanılabilir.



Resim 1.2 : MSP430 Launch Pad'i USB porta taktığımız an, Windows aygıt yöneticisindeki yeri.

Artık erişebileceğimiz kontrol edebileceğimiz bir kontrolörü bilgisayar sistemine tanıtmış olduk. Bu kontrolör ile üzerinde hazır bulunan sıcaklık sensörünün ürettiği bilgiyi, görsel programlama ile elde edip, sonucu bir text dosyasına kaydetmek, sonrasında da bu dosyayı web'e aktarma işlemleri olacaktır. Tüm bu işlemlerin sonunda **Sıcaklık Okuma Otomasyon Sistemi** adlı projemizi de bu modül sonuna kadar bitirmiş olursunuz.

1.2. Sistem Kontrolörünün programlanması

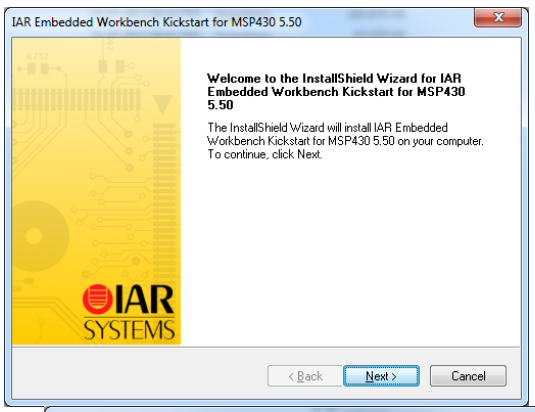
Mikro denetleyicilerin, donanımına erişmek için en uygun programlama dili C veya C++ dilidir. Günümüzde bir çok yerli ve yabancı üniversiteler, yazılım ve donanım üreticileri, mikro denetleyici ile sistemi kontrol etmek amacıyla C programlama dilini kullanmaktadırlar.

MSP430 Launch Pad geliştirme aracımızda IAR SYSTEMS yazılım geliştirme araçlarından IAR Embedded Workbench for TI MSP430 geliştirme yazılımını kullanacağız. Bu yazılımın özellikleri;

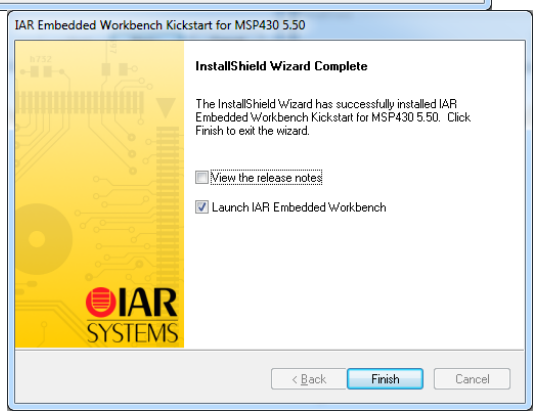
- C ve C++ compiler,
- FET debugger desteği,
- Run-time kütüphaneleri,

- MSP430 için assembler, Linker ve kütüphane araçları,
- PDF formatında, kullanıcı ve referans rehberleri,
- İşlemciye göre süre sınırlamasız 4 KB veya 8 KB ücretsiz kullanım izni.

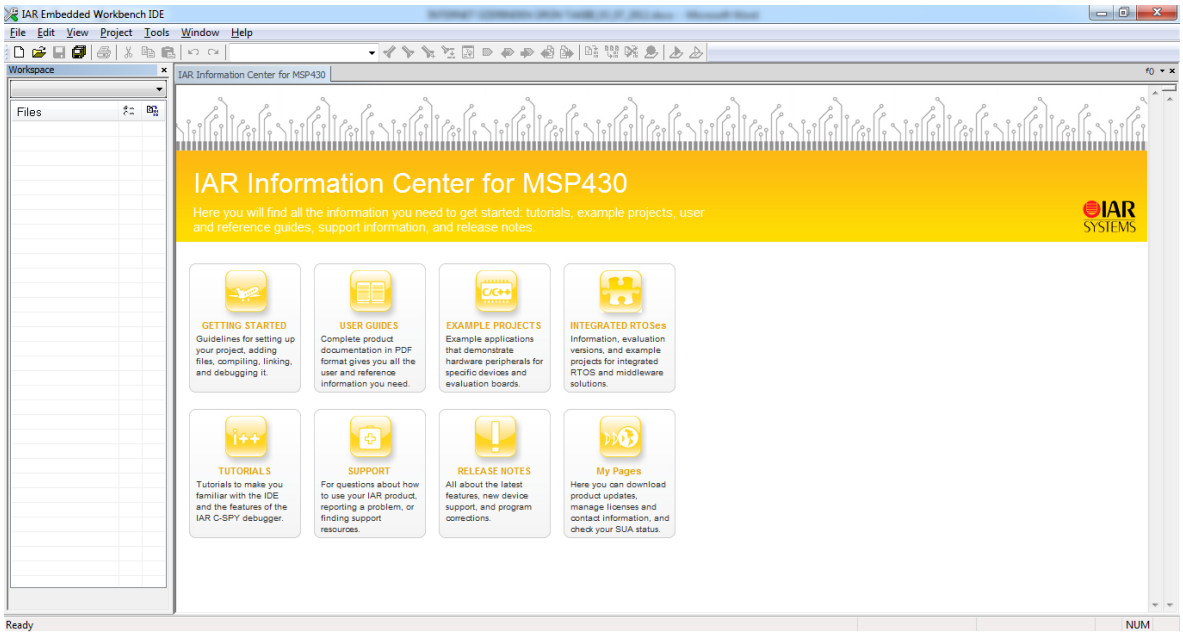
Resmi web sitesinden indirilen IAR programını indirme sayfasından, yazılımı bilgisayarınıza indirip, yükleyiniz (4 KB kod sınırlı ücretsiz). Ekrana gelen yönergeleri dikkatlice inceleyerek next (ileri) butonları ile yazılımın yüklenmesini bekleyiniz.



Resim 1.3 : IAR yükleme ekranı

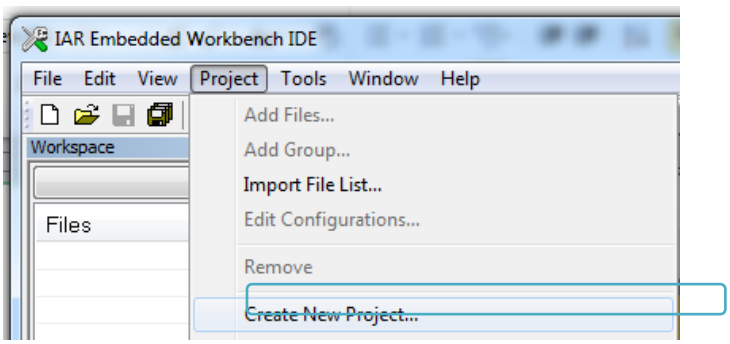


Resim 1.4 : IAR yükleme işlemi tamamlandı



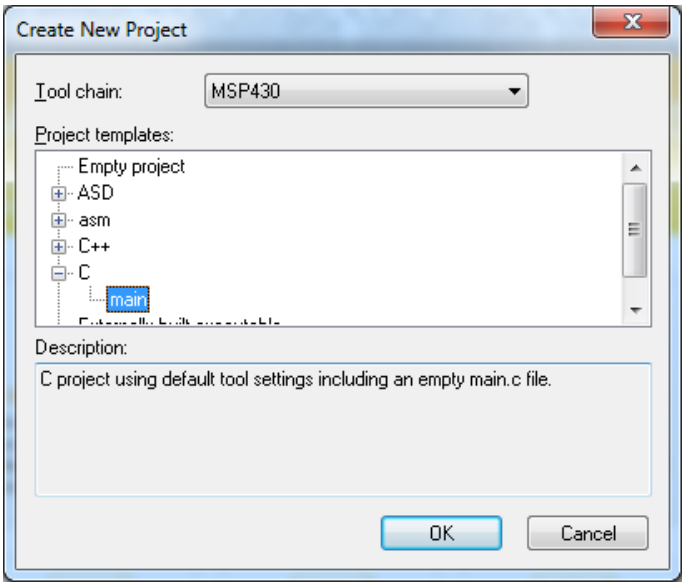
Resim 1.5 : IAR yazılımına ait çalışma anı ekran görüntüsü

Boş bir MSP430 projesi oluşturmak için;



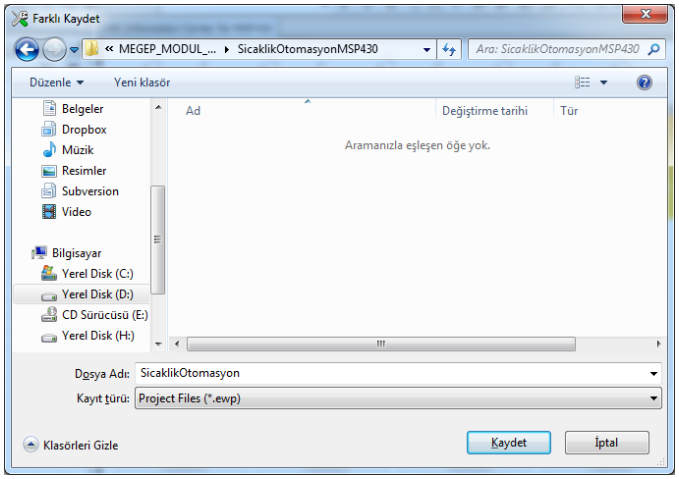
Resim 1.6 : Yeni IAR proje oluşturma ekran görüntüsü

IAR Embedded Workbench IDE yazılımının Project menüsünden Create New Project (yeni proje oluştur) komutu seçilir.

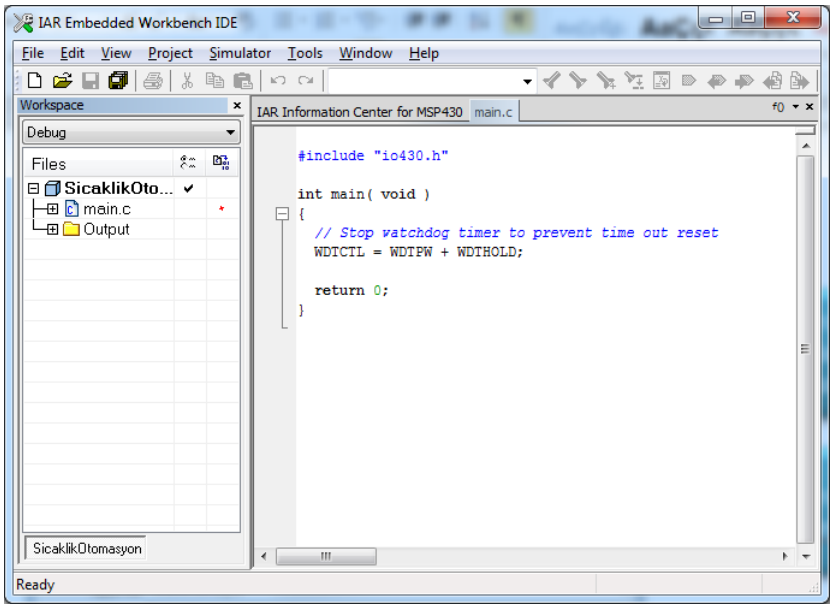


Resim 1.7 : Proje şablon ekranı

Tool chain (araç tipi) açılır kutusundan MSP430 seçilir. **Project templates** (proje kalıpları) penceresinden programlama dili seçimi yapılır ve **main.c** isminde hazır bir dosya üretilmiş olarak karşımıza gelmesi sağlanır. OK (Tamam) butonu ile proje oluşturulur.

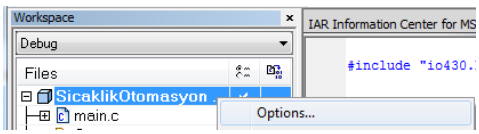


Resim 1.8 : MSP430 mikro denetleyicisi için IAR yazılımında proje oluşturma aşamaları

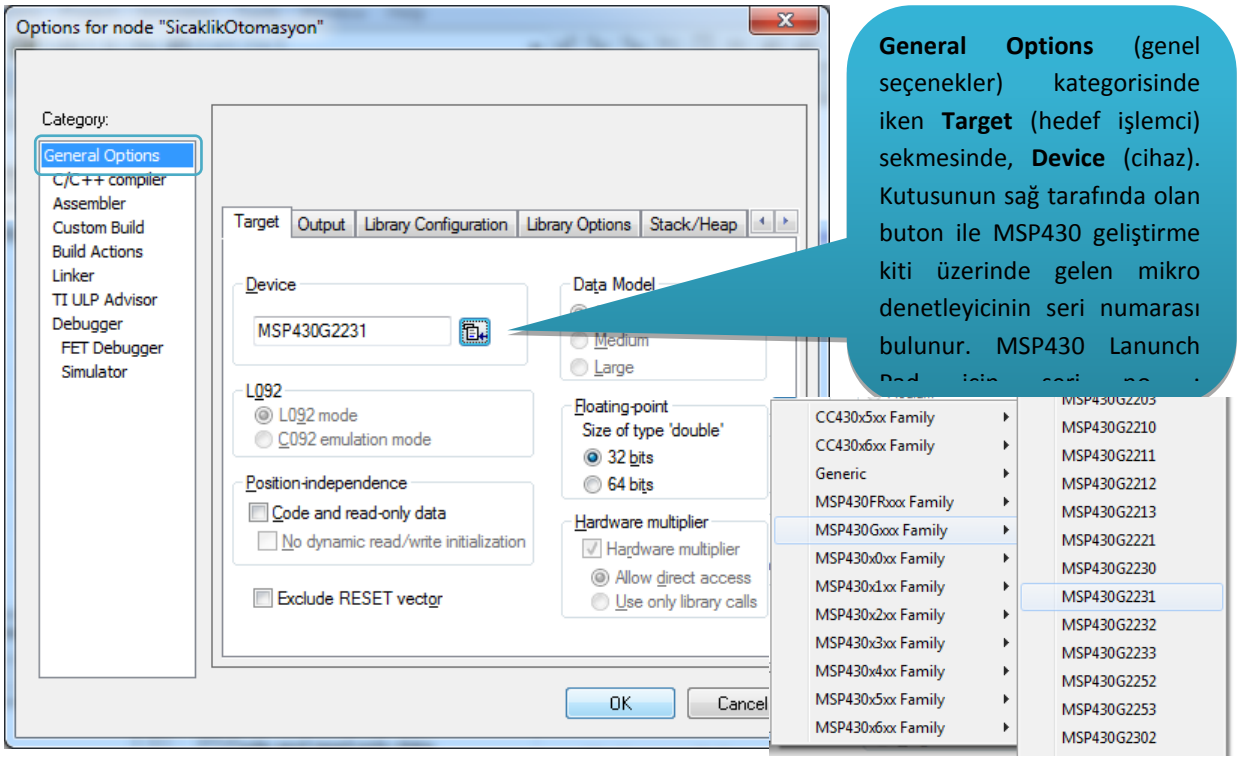


Resim 1.9 : MSP430 mikro denetleyicisi için boş bir proje karşımıza gelir

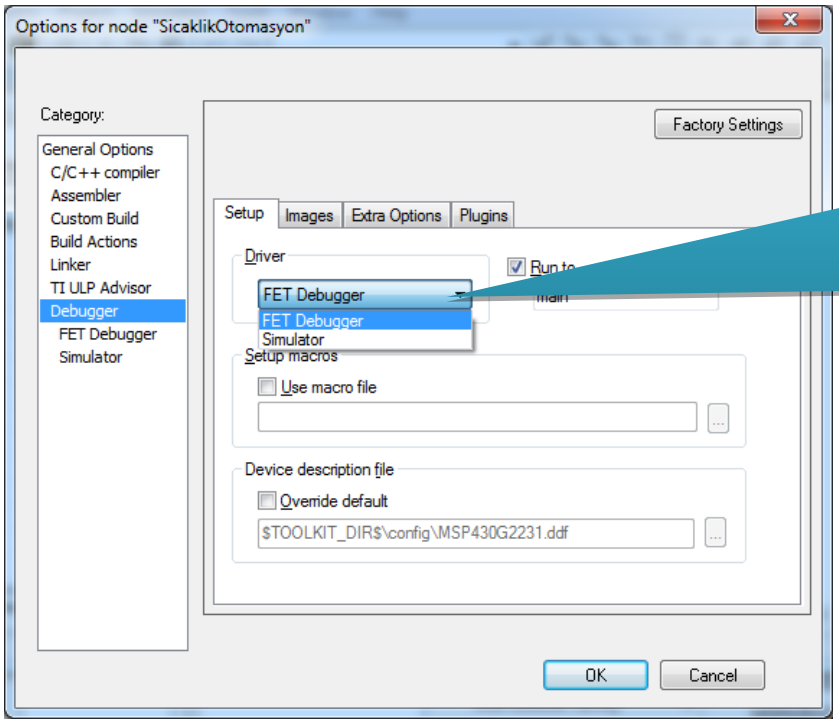
Oluşturulan boş MSP430 projesinde, ilk yapılması gereken işlemler vardır. Bunlar, workspace (çalışma alanı) penceresinde olan proje ismimizin üzerine sağ fare tuşu ile tıkladığımızda menü ekranı gelir. Gelen menüden **options** (seçenekler) menüsü seçilir. Karşımıza gelen pencereden, en önemli iki olay çok dikkatlice yapılması gerekmektedir.



Resim 1.10 : Proje ismi farenin sağ tuşu ile tıklanınca seçenekler menüsü gelir



Resim 1.11: Kullanılacak işlemci seçim ekranı



Kategori penceresinden **Debugger** (hata ayıklayıcı) fare ile tıklanarak seçilir. **Setup** (kurulum) sekmesinden **Driver** (sürücü) açılır kutusundan **FET Debugger** seçilir. Bu seçenek MSP430 Launch

Resim 1.12 : Bilgisayara bağlı olan MSP430 Launch Pad'e direk erişim sağlamak için FET Debugger seçilmelidir

OK (tamam) butonu ile projemize ait seçenekler penceresinden çıkılır. Yapılan işlemlerden sonra proje muhakkak kaydedilmelidir. Artık projenizi başka bilgisayarlara veya klasörlere bile taşısanız bu ayarlar her yerde geçerlidir.

MSP430 Launch Pad'in üzerinde hazır gelen sıcaklık sensörüne C programlama dili ile erişip, o bilgiyi seri port haberleşme ile görsel programalama dili ile oluşturacağımız yazılımımıza gönderme işlemine başlayabiliriz. IAR projesindeki **main.c** isimli dosyaya **sıcaklık okuma otomasyon sisteminde** kullanılacak olan kodları yazalım.

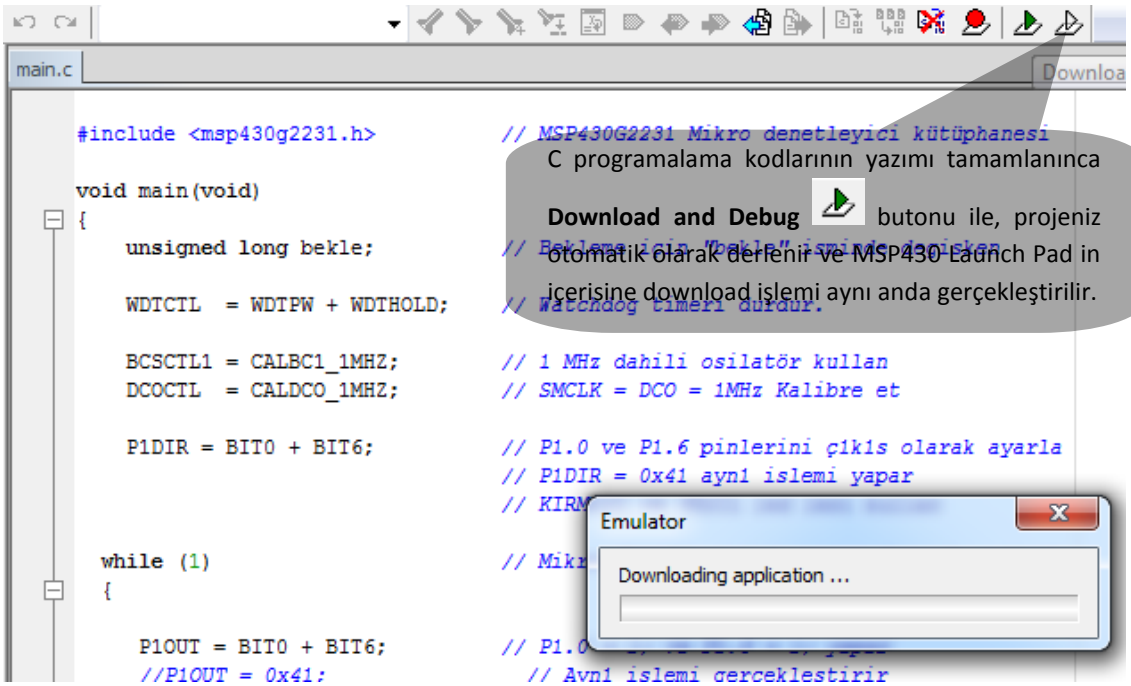
MSP430 Launch Pad ile C programlama dilinin en temel kalıbı ve iki adet led çalışma örneği şöyledir;

```c
#include <msp430g2231.h>

void main(void)
{

  while(1)
  {

  }

}
```

Resim1.13 : MSP430G2231 mikro denetleyicileri için IAR ile oluşturulmuş C kalıbı

```c
#include <msp430g2231.h>          // MSP430G2231 Mikro denetleyici kütüphanesi

void main(void)
{
    unsigned long bekle;          // Bekleme için "bekle" isminde degisken

    WDTCTL  = WDTPW + WDTHOLD;    // Watchdog timeri durdur.

    BCSCTL1 = CALBC1_1MHZ;        // 1 MHz dahili osilatör kullan
    DCOCTL  = CALDCO_1MHZ;        // SMCLK = DCO = 1MHz Kalibre et

    P1DIR = BIT0 + BIT6;          // P1.0 ve P1.6 pinlerini çikis olarak ayarla
                                  // P1DIR = 0x41 aynı islemi yapar
                                  // KIRMIZI ve YESIL led leri kullan

  while (1)                       // Mikro denetleyiciyi sonsuz dongude çalıstır
  {

     P1OUT = BIT0 + BIT6;         // P1.0 = 1; ve P1.6 = 1; yapar
     //P1OUT = 0x41;              // Aynı islemi gerçeklestirir

     bekle = 35000;               // bekle degiskenine 35000 degeri atanır.
     do (bekle --);               // bekle = bekle - 1; yapar
     while (bekle != 0);          // bekle <> 0; dongu içi çalısır.

     P1OUT = !BIT0 + !BIT6;       // P1.0 = 0; ve P1.6 = 0; yapar
     //P1OUT = 0x00;              // Aynı islemi gerçeklestirir

     bekle = 35000;               // bekle degiskenine 35000 degeri atanır.
     do (bekle --);               // bekle = bekle - 1; yapar
     while (bekle != 0);          // bekle <> 0; dongu içi çalısır.

  }
}
```

Resim 1.14 : MSP430 Launch Pad üzerinde bulunan KIRMIZI ve YEŞİL ledleri yakan C kodları



Resim 1.15: C kodlarını yazım işlemi tamamlanınca,
USB portuna bağlı olan MSP430 a derlenmiş dosyanın gönderilmesi işlemi

MSP430 Launch Pad üzerinde hazır gelen sıcaklık sensöründen bilgi okumak için, boş bir IAR projesi başlatılır. İlk satırlarına dâhil edilecek **„io430x22x2.h“** kütüphanesi, gönderilecek bilgi tanımlaması ile 9600 baud için 104 değeri (1 MHZ / 9600) tanımlanır. Sıcaklık bilgileri için kullanılacak tamsayı ve ondalıklı sayı değişkenleri tanımlanır. Gömülü sistem projemizde iki adet rutinimiz olacaktır. Birincisi ana (**main**) rutin (procedure) diğeri ise okunan sıcaklık değerini kesmelere bağlı olarak bilgisayarımıza seri portta gönderecek olan iletim rutini (**Transmit**).

Ana prosüdürümüzde (rutin) power kesme problemlerini önleyen WatchDog Timer durdurulur. 1 MHZ lik dâlilî osilatör seçilim osilasyon kalibresi yaklaşık %1 veya %2 hata payı olacak şekilde set edilir. ADC10 isimli ADC modülünün kanal seçimi ve bölücüsü ayarlanır. Referans seçimi yapıldıktan sonra ASC10ON ile aktif edilir. Veri gönderme pini çıkış (mikro denetleyiciden pc ye bilgi gidecek) olarak ayarlanır. Kesmelere (Interrupt) izin verilir. Ana rutinde buraya kadar olan işlemler, MSP430'a enerji verildikten sonra veya reset butonuna basıldıktan sonra sadece bir kere çalışır. Ana rutinde sonsuz döngü (**while (1)**) içerisinde ise; ADC10 kendi osilasyon kaynağını (ADC10SC) kullanır. Her bir ADC10 çeviriminde kesme olup olmadığına bakılır. Transmit rutini çıkışa bilgi (TX) gönderdiği için otomatik olarak kesme üretilmiş olur. Dolaysıyla ADC10IFG bu durumu algılar ve uygular. Bir sonraki kesme işlemine hazır olması için eski kesme içeriğini karar yapısı içerisinde temizler. Ham işlenmemiş veri manasına gelen **ham** isimli değişkene MSP430, sürekli hazır olan ve enerjilendiği zaman otomatik olarak dâlilî ısı sensöründen aldığı bilgiyi

ADC10MEM isimli fonksiyonda saklayan bilgiyi atar. Atanan bilgi artık ikinci rutinimiz olan **Transmit**() çalıştırılmasıyla ana rutinin işi biter. Ama sonsuz döngü olduğu için işlemler aynı şekilde (1 MHZ) devam eder.

İkinci rutinimiz transmit (iletim) kısmındaki olaylarımız ise; TXByte isimli değişkene dur bitini (stop bit) 10 byte'lık bir veriyi pc ye ileteceği, ilk zamanını nasıl anlayacağı bilgilerini işler ve önceki bilgiler gitmeden yeni bir gönderme işlemi yapmaz.

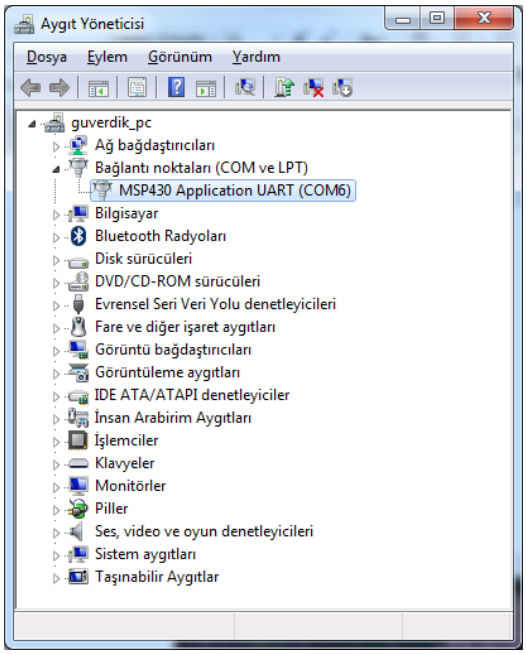
IAR Information Center for MSP430 **main.c**

```
#include <io430x22x2.h>

#define TXD       0x02
#define Bit_Zaman 104            // 9600 Baud, SMCLK=1MHz (1MHz/9600) = 104
#define OUTMOD2   0x80
#define CCIS0     0x1000

unsigned char BitSayac;          // Bit sayaç değişkeni
unsigned int TXByte;             // veri gönderim değişkeni
long ham, derece;                // Sıcaklık ve A/D çevrim sonuç değişkeni

void Transmit(void);

void main (void)
{
      WDTCTL = WDTPW + WDTHOLD;   // Watchdog timeri durdur
      BCSCTL1 = CALBC1_1MHZ;      // 1 MHz dahili osilatör
      DCOCTL = CALDCO_1MHZ;       // SMCLK = DCO = 1MHz kalibreli

      ADC10CTL1 = INCH_10 + ADC10DIV_3;  //ADC10 10. kanal kullanılıyor.
      ADC10CTL0 = SREF 1 + ADC10SHT 3 + REFON + ADC10ON ; //ADC10 Aktif (ON)

      P1SEL |= TXD;              // Veri gönderme pin ayarı
      P1DIR |= TXD;              // TXD, Veri göndermeyi ÇIKIS yap

      __bis_SR_register(GIE);    // Kesmelere izin ver

  while (1)           // Sonsuz döngü
  {
    ADC10CTL0 |= ENC + ADC10SC;       // ADC10 kendisinin Clock kullan
    if(ADC10CTL0 & ADC10IFG)          // ADC10 çevrim (kesme var mı?)
    {
      ADC10CTL0 &= !ADC10IFG;        // ADC10 kesme temizle
      ham = ADC10MEM;                // ADC10 içersindeki saklayıcıdan oku
      derece = ((ham - 673) * 423) / 1024;    //Sıcaklığı dereceye çevir.
      TXByte = derece;                        // Hesaplananı TXByte ata
      Transmit();                             // Okunan sıcaklığı gönder.
    }
  }
}


//*** BU FONKSIYON TXByte DEGISKENINDEKI VERIYI TXD PININDEN GONDERIR  *******
void Transmit(void)
{
  TXByte |= 0x100;          // Dur bitini ekle TXByte (lojik 1)
  TXByte = TXByte << 1;     // Başlangıç bitini ekle(lojik 0)
  BitSayac = 0xA;           // Gönderilecek bit sayısını yükle (10 Byte)
                            // (Toplam 10 bit = 8 data + Başlangıç + stop)
  CCTL0 = OUT;              // Txd hattını boşa al
  TACTL = TASSEL_2 + MC_2; // SMCLK, sürekli mod
  CCR0 = TAR;               // Karşılaştırıcı değişkenini düzenle
  CCR0 += Bit_Zaman;        // İlk bit zamanını ayarla
  CCTL0 = CCIS0 + OUTMOD0 + CCIE; //Karşılaştırıcı kontrol değişkenini düzenle
  while ( CCTL0 & CCIE );  // Önceki gönderme tamamlanana kadar bekle
}
```

Resim 1.16 : MSP430 Sıcaklık Okuma Otomasyon Sistemi için gömülü sistem kodlarının tamamı

1.3. Programlama Kontrolör Bağlantısı

Gömülü sistem yazılımı gerçekleştirilmiş kontrolörümüze (kontrol kartı – MSP430 Launch Pad) görsel programlama ile erişimi gerçekleştirmek için, windows aygıt yöneticisindeki MSP430 kartına windows tarafından atanan COM numarasını öğrenmemiz gerekir.



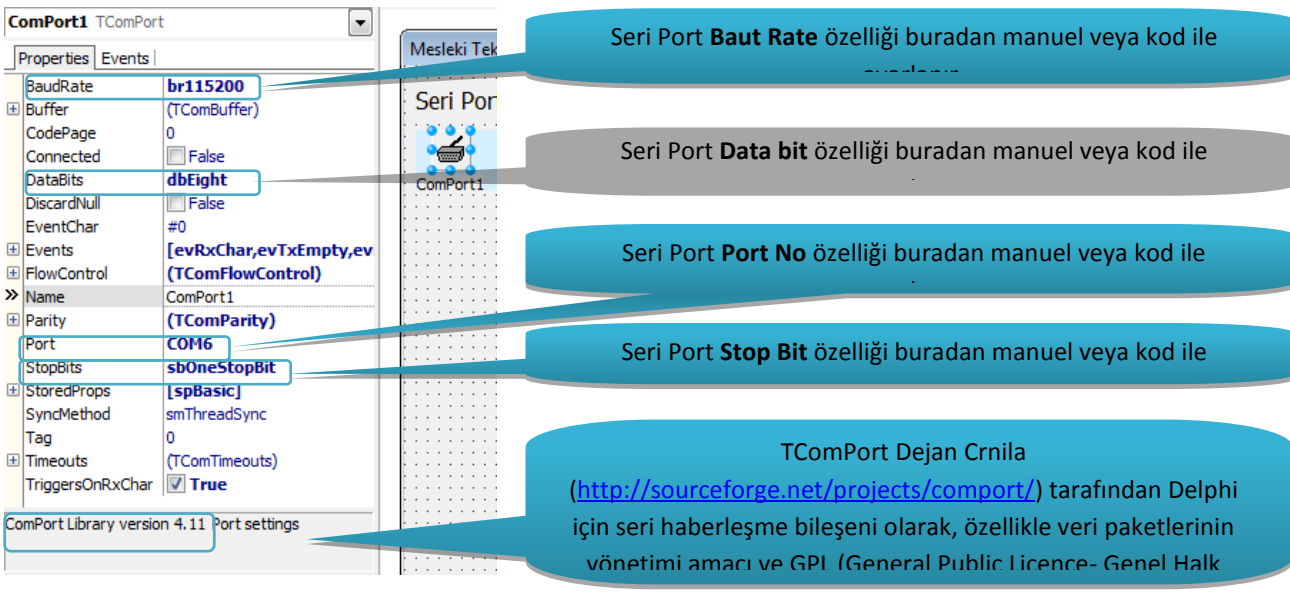
Resim 1.17 Windows tarafından MSP430 Launch Pad'e atanan seri port no

Görsel programlama dili boş bir windows proje (application) oluşturup form üzerine aşağıdaki görüntüde olduğu gibi gerekli bileşenleri (component) getirip, özellikleri ayarlanır.

Resim 1.18 : Görsel programlama ile oluşturulmuş form tasarımı

Kontrol kartına erişmek için gerçekleştirilen form tasarımında en önemli unsur ComPort1 bileşenidir. Diğer bileşenler (component) etiketler (Label), butonlar (Button) ve

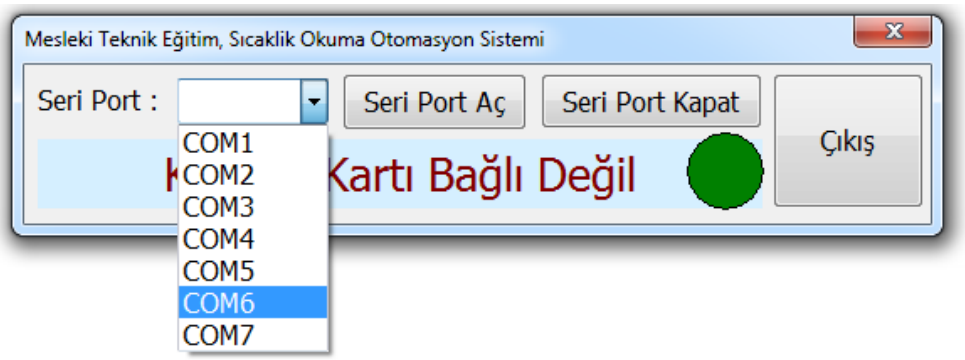
kontrol kartına bağlı olup olmadığımızı anlamamızı görsel olarak algılamamızı sağlayan biçim (Shape) bileşenlerinden oluşmaktadır. Seri port bileşenine (component) ait özelliklerde en önemli dört nokta vardır ve bu özellikler tüm programlama dillerinde de aynıdır. Bunlar Seri Port No (**COM6**), Baut Rate (iletim hızı **115200**), Stop Bit veya Data Bit (dur biti veya veri biti **0**), ve eşlik bitidir (parity **none**).



Resim 1.19 : Delphi XE2 içerisine yüklenmiş olan TComPort bileşenine ait özellikler (properties).

Resim 1.20 : Visual Studio 2010 Visual Basic formuna hazır gelen Serial Port bileşeninin özellikleri (properties)

Form tasarımındaki açılır kutudan hangi port no kullanılacak ise seçilir ve **Seri Port Aç** butonu ile seri iletişim gerçekleştirilir. **Seri Port Aç** butonunun **Click** (tıklama) olayına ise; açılır kutudan herhangi bir port seçilmiş mi, değil ise uyarı mesajı gelsin, Label bileşeninde bağlandığına dair mesaj gösterilsin, daire biçimde (circle shape) ise yeşil renk, açık yeşil renke dönüştürülsün işlemleri gerçekleştirilir.



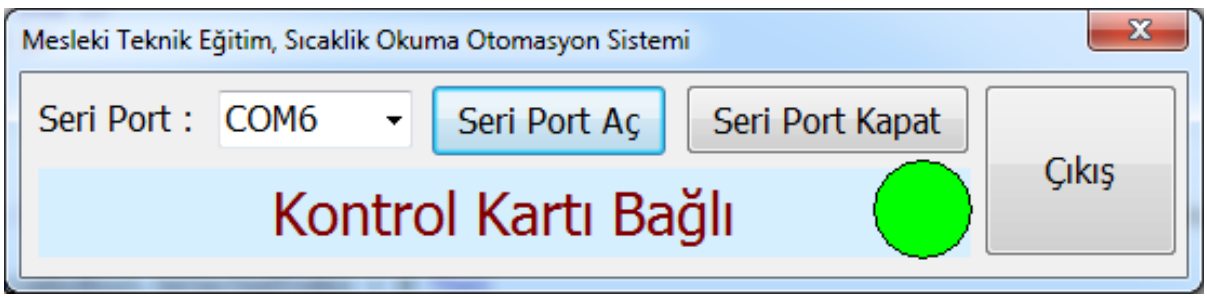
Resim 1.21 : Çalışma anında açılır kutunun durumu

```
procedure TForm3.Button2Click(Sender: TObject);
begin

  if ComboBox1.ItemIndex < 0 then
     begin
       ShowMessage('Lütfen Seri Port No Seçiniz');
       Exit;
     end;

  Label2.Caption := 'Kontrol Kartı Bağlı';
  Shape1.Brush.Color := clLime;
  ComPort1.Port := ComboBox1.Text;
  ComPort1.Connected := True;

end;
```

Resim 1.22 : Seri Port Aç butonuna ait Click (tıklama) olaylarının kodları (Delphi XE2)

```
Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click

    If ComboBox1.SelectedIndex < 0 Then
        MsgBox("Lütfen Seri Port No Seçiniz")
        Exit Sub
    End If

    Label2.Text = "Kontrol Kartı Bağlı"
    OvalShape1.FillColor = Color.Lime
    SerialPort1.PortName = ComboBox1.Text
    SerialPort1.Open()

End Sub
```

Resim 1.23 : Seri Port Aç butonuna ait Click (tıklama) olaylarının kodları (VS2010 Visual Basic)



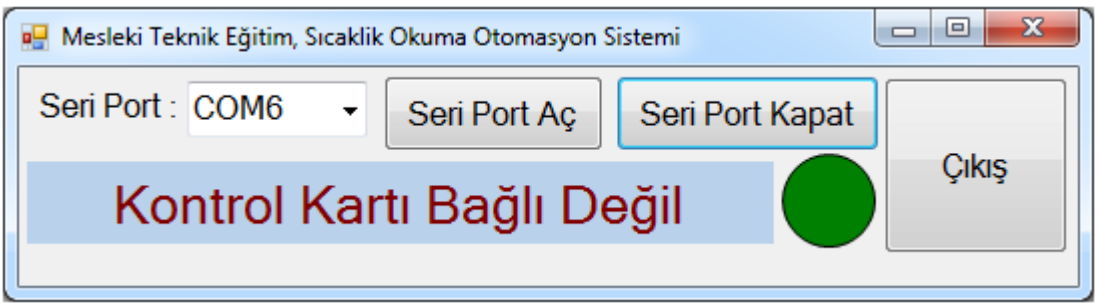
Resim 1.24: Seri iletişim ile MSP430 Launch Pad kontrol kartına bağlı olduğuna dair çalışma anı (runtime) ekranı

```
procedure TForm3.Button3Click(Sender: TObject);
begin

  Label2.Caption := 'Kontrol Kartı Bağlı Değil';
  Shape1.Brush.Color := clGreen;
  ComPort1.Connected := False;

end;
```

Resim 1.25 : Seri Port Kapat butonuna ait Click (tıklama) olayına ait kodlar (Delphi XE2)

```
Private Sub Button3_Click(sender As System.Object, e As System.EventArgs) Handles Button3.Click

    Label2.Text = "Kontrol Kartı Bağlı Değil"
    OvalShape1.FillColor = Color.Green
    SerialPort1.Close()

End Sub
```

Resim 1.26 : Seri Port Kapat butonuna ait Click (tıklama) olayına ait kodlar (VS2010 Visual Basic)



Resim 1.27 : Seri Port Kapat butonu çalıştırıldıktan sonra elde edilen çalışma anı (runtime) ekran görüntüsü

Projemize ait çıkış butonuna ait kodlarımız ise şöyledir.

```
procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin

  if Application.MessageBox('Programdan Çıkmak İstiyormusunuz?',
  'Programdan Çıkış', mb_YesNo) = IdYes then
  Halt else Abort;

end;
```

Resim 1.28 : Çıkış butonuna ait Click (tıklama) olayına ait kodlar (Delphi XE2)

```
Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

    If MsgBox("Programdan Çıkmak İstiyormusunuz", MsgBoxStyle.YesNo,
              "Programdan Çıkış") = MsgBoxResult.Yes Then
        Close()
    End If

End Sub
```

Resim 1.29 : Çıkış butonuna ait Click (tıklama) olayına ait kodlar (VS2010 Visual Basic)

1.4. Kontrolörden Alınan Verilerin Text Tipi Dosyaya Kaydedilmesi

Görsel programlama ile seri iletişim bileşenlerini (component) kullanarak kontrol kartımız ile bağlantımızı gerçekleştirdik. Bu aşamadan sonra gömülü sistem tarafından gönderilen **TXByte** (**Transmit Byte – gönderilen byte**) değişkenindeki bilgiyi artık seri porttan alıp inceleyebiliriz daha sonra TEXT tipindeki bir dosyaya kaydını gerçekleştirip, bu dosyayı da istediğimiz bir klasöre kaydedebiliriz. Tüm işlemlerimizin çalıştırıldığı bilgisayarda **web-server** windows bileşeninin (özelliğinin) açık olduğunu muhakkak denetleyin. Bu işlemlerden sonra TEXT tipindeki dosyadan veri okuyup web sayfasından gösterme olaylarını gerçekleştireceğiz. Seri port bileşenindeki **OnRxChar** (bir karakter al – oku) olayına (**events**), global olarak tanımlanan **GelenHamBilgi** değişkenine mikro denetleyicinin (kontrol kartı) **TXByte** isminde göndermiş olduğu değer atanır.

```
procedure TForm3.ComPort1RxChar(Sender: TObject; Count: Integer);
begin

  ComPort1.ReadStr(GelenHamBilgi, Count);

end;
```

```
public
  { Public declarations }
  GelenHamBilgi : String;
```

Resim 1.30 : Kontrol kartından seri port ilen gelen bilgiyi, görsel programdaki TComPort bileşeni vasıtasıyla bilgi yakalanır ve global (heryerde kullanılabilen değişken) olarak String (Yazı – Metin) tipinde tanımlanmış değişkene değer atama işlemi (Delphi XE2)

```
Private Sub SerialPort1_DataReceived(sender As System.Object, _
            e As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived

    GelenHamBilgi = SerialPort1.ReadByte()

End Sub
```

```
Public Class Form1

    Public GelenHamBilgi As Integer = 0
```

Resim 1.31 : Kontrol kartından seri port ilen gelen bilgiyi, görsel programdaki SerialPort bileşeni vasıtasıyla bilgi yakalanır ve global (her yerde kullanılabilen değişken) olarak Integer (Tam sayı) tipinde tanımlanmış değişkene değer atama işlemi (VS2010 Visual Basic)

Gelen bilgiyi Timer bileşenimizle

analiz işlemlerimizi gerçekleştirip etiketimizde (label) görsel hâle dönüştürme işlemlerine geçilir. Bu Timer içerisinde olayları aynı zamanda TEXT dosyamıza ne kadar sn aralıklarla dosya güncellemesi olacak, işlemlerinde de ayrıca kullanacağız.

```
procedure TForm3.Timer1Timer(Sender: TObject);
var I, RXData  :  Integer;
begin

  for I := 1 to Length(GelenHamBilgi) do
     begin
        RXData := StrToInt('0x' + IntToHex(Ord(GelenHamBilgi[I]), 1));
        Label4.Caption := IntToStr(RXData) + '°C';
     end;

end;
```

Resim 1.32 : Timer olayında GelenHamBilgi (aynı zamanda string dizidir) değişkenindeki değerin ASCII karakter kod bilgisi elde ediliyor. Elde edilen Tamsayı (Integer) bilgisi Hex koduna çevriliyor. Tekrar RXData Integer tipindeki değişkene atanıp bu değişken içeriği Label da görsel hale devamına °C eklenerek dönüştürülüyor. (Delphi XE2)

```
Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs) Handles Timer1.Tick

    Label4.Text = GelenHamBilgi.ToString + "°C"

End Sub
```

Resim 1.33 : Timer olayında GelenHamBilgi (integer) değişkenindeki değer yazıya (string) dönüştürülüp °C eklenerek Label bileşenine aktarılıyor (VS2010 Visual Basic)

Resim 1.34 : Timer bileşeni aktif edildiği (Timer1.Enable = True "properties") zaman kontrol kartının üzerinde bulunan sıcaklık sensöründen elde edilen bilgi label bileşeninde görsel hale dönüştürüldü

Kontrol kartından okuduğumuz bilgiyi TEXT dosyasına kaydetme aşamasında, dosyaya kayıt ne sıklıkta olacak ve dosya hangi konuma (sürücü, klasör yolu) kaydedilecek bilgilerine karar verip uygulama aşamasına geçelim. Kayıt sıklığını açılır kutu ile **2,3,4 sn** aralıklarla dosyaya kaydetsin ve dosya yolu ve dosya adı da **C:\inetpub\wwwroot\sicaklik\sonuc.txt** olsun. **C:\inetpub\wwwroot** bu kısım windows web-server'a aittir. **\sicaklik** kısmı ise sanal klasördür (virtual folder). Elde ettiğimiz ve dosyaya kaydını yaptığımız bilgimizi web te yayınlamak için ayrıca bir web perojesi (WebApplication) yapacağız ve o uygulamamızın adı **sicaklik** olacaktır. Form tasarım ekranımızda text kutusu ve kayıt sıklığı için açılır kutu (ComboBox) ve kayıt başlatmak veya durdurmak için işaret kutusu (CheckBox) vardır. **Timer1** (zamanlayıcı 1) bileşeninin **Timer** (VB için **Tick**) olayına ilave kod olarak; **Dosyaya Kaydı Başlat** (CheckBox1 bileşeni) işaret kutulu bileşen ile dosyaya istediğimiz zaman kaydetme işlemini başlatıp durdurma olayını gerçekleştirebiliriz.

Resim 1.35 : Text dosyasına kaydetme sıklığı, Dosyaya kayıt başlama ve durdurma ile kayıt yerini gösterir ekran görüntüsü

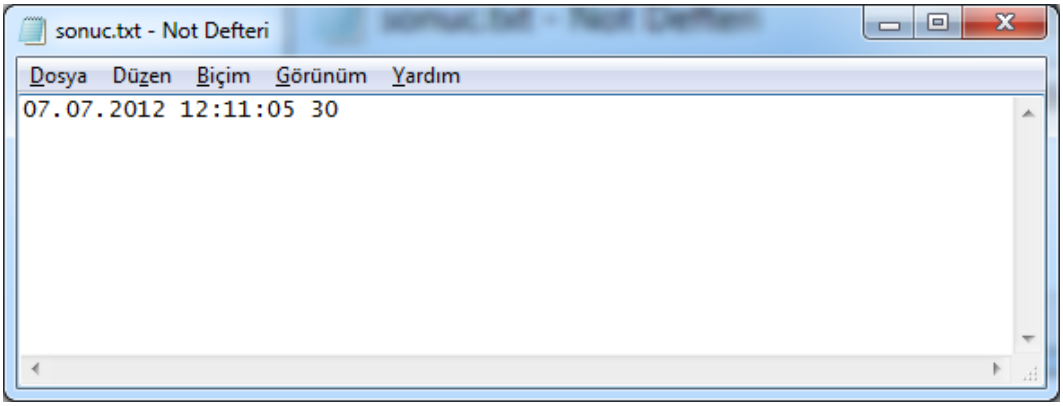
```
procedure TForm3.Timer1Timer(Sender: TObject);
var I, RXData  :  Integer;
    SonucDosyasi : TextFile; // Text File (text dosyası) tipinde değişken üret
begin

  for I := 1 to Length(GelenHamBilgi) do
     begin
        RXData := StrToInt('0x' + IntToHex(Ord(GelenHamBilgi[I]), 1));
        Label4.Caption := IntToStr(RXData) + '°C';
     end;

  //************  DOSYAYA BİLGİ KAYDETME İŞLEM BAŞLANGICI  ********************
  if CheckBox1.Checked then
     begin

        // Hard diskimizdeki dosyayı TextFile tipindeki değişkenimize ata
        AssignFile(SonucDosyasi, Edit1.Text);

        // yazma (write) amaçlı ata
        ReWrite(SonucDosyasi);

        //Dosyamıza bilgilerin (zaman ve sıcaklık) kaydedilmesini gerçekleştir
        WriteLn(SonucDosyasi, DateTimeToStr(Now) + ' ' +  IntToStr(RXData));

        // Dosyayı kapat
        CloseFile(SonucDosyasi);

     end;
  //***************  DOSYAYA BİLGİ KAYDETME İŞLEM SONU   **********************

end;
```

Resim 1.36 : Timer bileşeninin Timer olayına ilave kodlarla zaman ve sıcaklığı ilgili text dosyasına kayıt işlemi
(Delphi XE2)

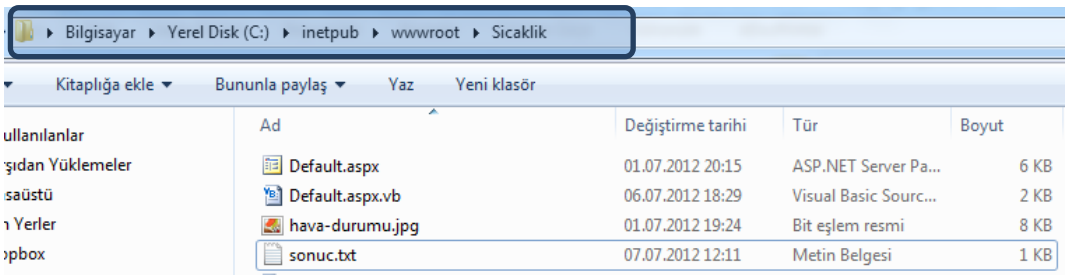
```
Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs) Handles Timer1.Tick

    Label4.Text = GelenHamBilgi.ToString + "°C"


    '************  DOSYAYA BİLGİ KAYDETME İŞLEM BAŞLANGICI  ********************
    If CheckBox1.Checked Then

        ' Hard diskimizdeki dosyayı TextFile tipindeki değişkenimize ata
        Dim SonucDosyasi As New System.IO.StreamWriter(TextBox1.Text)

        'Dosyamıza bilgilerin (zaman ve sıcaklık) kaydedilmesini gerçekleştir
        SonucDosyasi.WriteLine(System.DateTime.Now.ToString + " " + GelenHamBilgi.ToString)

        ' Dosyayı kapat
        SonucDosyasi.Close()

    End If
    '***************  DOSYAYA BİLGİ KAYDETME İŞLEM SONU   **********************

End Sub
```

Resim 1.37 : Timer bileşeninin Tick olayına ilave kodlarla zaman ve sıcaklığı ilgili text dosyasına kayıt işlemi
(VS2010 Visual Basic)

Resim 1.38 : Çalışma anında (runtime) kontrol kartından seri port aracılığı ile elde edilen sıcaklık bilgisi etikette (Label) gösteriliyor. Dosyaya Kaydı Başlat işaret kutusu (CheckBox) seçilince zaman ve sıcaklık bilgisi TextBox kutusundaki sürücü, klasör ve dosyaya yazma işlemi gerçekleştiriliyor



Resim 1.39 : Zaman ve sıcaklık bilgisi kaydedilmiş text dosyanın içeriği



Resim 1.40 : Zaman ve sıcaklık bilgisi kaydedilmiş dosyanın sürücü ve klasör konumunu gösterir ekran görüntüsü

UYGULAMA FAALİYETİ

MSP430 Launch Pad ile Led Yakıp Söndürme Uygulaması

İşlem Basamakları
1. MSP430 Launch Pad i USB porta takınız.
2. Windows aygıt yöneticisinden COM bağlantısının gerçekleşip, gerçekleşmediğini denetleyiniz.
3. Boş bir IAR projesi oluşturunuz.
4. IAR seçenekler (**Options**) kısmından Device (**MSP430G2231**) işlemci ile Debugger dan **FET debugger** seçimini gerçekleştiriniz.
5. **main.c** sayfasına gerekli c kodlarını yazınız ve mikro denetleyiciye download işlemini gerçekleştiriniz. Kırmızı ve yeşil ledlerin kısa süreli aralıklarla yanıp söndüklerini gözlemleyiniz.

```
#include <msp430g2231.h>          // MSP430G2231 Mikro denetleyici kütüphanesi

void main(void)
{
    unsigned long bekle;          // Bekleme için "bekle" isminde degisken

    WDTCTL  = WDTPW + WDTHOLD;    // Watchdog timeri durdur.

    BCSCTL1 = CALBC1_1MHZ;        // 1 MHz dahili osilatör kullan
    DCOCTL  = CALDCO_1MHZ;        // SMCLK = DCO = 1MHz Kalibre et

    P1DIR = BIT0 + BIT6;          // P1.0 ve P1.6 pinlerini çikis olarak ayarla
                                  // P1DIR = 0x41 aynı islemi yapar
                                  // KIRMIZI ve YESIL led leri kullan

  while (1)                       // Mikro denetleyiciyi sonsuz dongude çalıstır
  {

     P1OUT = BIT0 + BIT6;         // P1.0 = 1; ve P1.6 = 1; yapar
     //P1OUT = 0x41;              // Aynı islemi gerçeklestirir

     bekle = 35000;               // bekle degiskenine 35000 degeri atanır.
     do (bekle --);               // bekle = bekle - 1; yapar
     while (bekle != 0);          // bekle <> 0; dongu içi çalısır.

     P1OUT = !BIT0 + !BIT6;       // P1.0 = 0; ve P1.6 = 0; yapar
     //P1OUT = 0x00;              // Aynı islemi gerçeklestirir

     bekle = 35000;               // bekle degiskenine 35000 degeri atanır.
     do (bekle --);               // bekle = bekle - 1; yapar
     while (bekle != 0);          // bekle <> 0; dongu içi çalısır.

  }
}
```

UYGULAMA FAALİYETİ

MSP430 Launch Pad ile Buton Kontrol Uygulaması

İşlem Basamakları
1. MSP430 Launch Pad'i USB porta takınız. Windows aygıt yöneticisinden COM bağlantısının gerçekleşip, gerçekleşmediğini denetleyiniz.
2. Boş bir IAR projesi oluşturunuz. IAR seçenekler (**Options**) kısmından Device (**MSP430G2231**) işlemci ile **Debugger** dan **FET debugger** seçimini gerçekleştiriniz.
3. **main.c** sayfasına gerekli c kodlarını yazınız ve mikro denetleyiciye download işlemini gerçekleştiriniz. Yeşil ledin kısa süreli aralıklarla yanıp söndüğünü, kullanıcı butonuna basıldığı zaman yeşil ledin söndüğünü, tekrar kullanıcı butonuna basıldığı zaman yeşil led in tekrar kısa süreli aralıklarla yanıp söndüğünü gözlemleyiniz.

```
#include <msp430g2231.h>

void main (void){

   unsigned char YANIP_SONME_DURUMU = 1;    // Led durumunu (YANIK/SÖNÜK)
   volatile unsigned int i;                 // döngü beklemeleri için
   WDTCTL = WDTPW | WDTHOLD;                // Watchdog Timer Durdur

   P1DIR |= 0x40;                           // P1.0 Çıkış olarak ayarla
   P1DIR &= ~0x08;                          // P1.3 Giriş olarak ayarla
   P1REN |= 0x08;                           // P1.3 Pull-Up/Down Direnci Aktif Et

   while(1){                                // Sonsuz Döngü..

     if (YANIP_SONME_DURUMU == 1) {
         P1OUT ^= 0x40;                     // P1.0 durumu 1 ise 0, 0 ise 1 yap
         i=15000; do (i--); while (i !=0); // döngü..
     }

     if (!(P1IN & 0x08)) {                  // Butona Basma durumunu Tara

       i=1500; do (i--); while (i !=0);     // Butana Basma süresi (döngü)
       while (!(P1IN & 0x08));              // Butona basmak için bekle
       i=1500; do (i--); while (i !=0);     // Butana Basma süresi (döngü)

       if (YANIP_SONME_DURUMU == 1){        // Eğer LED yanıp sönüyorsa durdur
         P1OUT &= ~ 0x40;                   // LED i söndür
         YANIP_SONME_DURUMU = 0;            // LED in SÖNÜK oldugunu unutma
       } else YANIP_SONME_DURUMU = 1;       // LED in YANIK oldugunu unutma
     }
   }
}
```

UYGULAMA FAALİYETİ

MSP430 Launch Pad ile PWM Kontrol Uygulaması

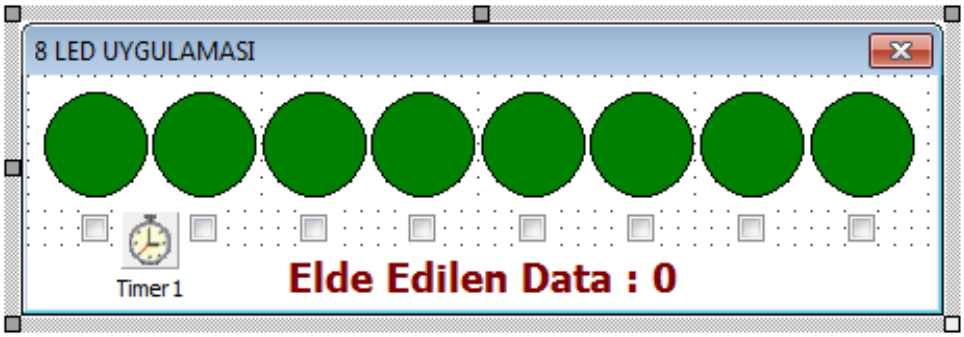
İşlem Basamakları
1. MSP430 Launch Pad'i USB porta takınız. Windows aygıt yöneticisinden COM bağlantısının gerçekleşip, gerçekleşmediğini denetleyiniz.
2. Boş bir IAR projesi oluşturunuz. IAR seçenekler (**Options**) kısmından Device (**MSP430G2231**) işlemci ile **Debugger** dan **FET debugger** seçimini gerçekleştiriniz.
3. **main.c** sayfasına gerekli c kodlarını yazınız ve mikro denetleyiciye download işlemini gerçekleştiriniz. Kod ile yazdığınız (1-98) arasında % cinsinden PWM değeri kadar Yeşil LED'in az ışık verdiğini (%10 pwm de) gözlemleyiniz. % pwm değerini değiştirerek yeşil LED'in parlaklığındaki değişimi de gözlemleyiniz.

```
#include <msp430g2231.h>

void main(void)
{
  WDTCTL = WDTPW + WDTHOLD;      // Watchdog Timer Durdur

  BCSCTL1 = CALBC1_1MHZ;         // 1 MHz dahili osilatör kullan
  DCOCTL  = CALDCO_1MHZ;         // SMCLK = DCO = 1MHz Kalibre et


  P1DIR |= BIT6;                 // P1.6 Çıkış
  P1SEL |= BIT6;                 // PWM Çıkışı P1.6 (YEŞİL led) Seçildi

  CCR0 = 99-1;                   // Periyot
  CCR1 = 10-1;                   // (PWM % si. "1-98 arası" %10) Duty Cy

  CCTL1 = OUTMOD_7;              // CCR1 reset/set modunda. Timer_A modü
                                 // bulunan Capture - Compare.
  TACTL = TASSEL_2 + MC_1;       // SMCLK, Up Mode

  _BIS_SR(LPM1_bits);            // SMCLK Kullanıldığı için LPM1. Kesme
}
```

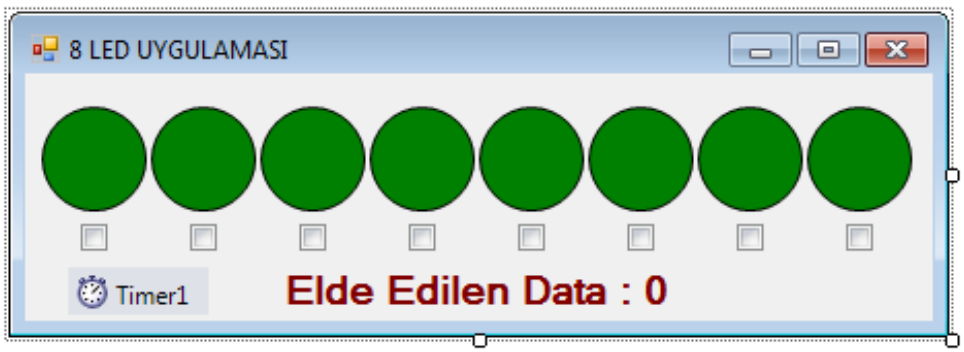
UYGULAMA FAALİYETİ

Görsel Programlama dili ile 8 bit veri elde etmek

İşlem Basamakları
1. Görsel Programlama dili ile boş bir Windows uygulaması (**application**) **oluşturunuz.**
2. Kod sayfasına, gerekli kodları yazıp, **elde edilen data** bilgisini gözlemleyiniz.
3. Aşağıdaki form yapısına gerekli bileşenleri getirip tasarımı gerçekleştiriniz.



Delphi XE2 Form Tasarımı. (Timer1 Interval 300, Enable = True)



VS2010 Visual Basic Form Tasarımı. (Timer1 Interval 300, Enable = True "properties")

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri **Evet**, kazanamadığınız becerileri **Hayır** kutucuğuna (**X**) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Sistemi hazırlayabildiniz mi?		
2. Sistem kontrolörünün programlayabildiniz mi?		
3. Görsel programlama dili ile kontrolöre erişebildiniz mi?		
4. Programlama ile kontrolör verilerini text dosyasına kaydedebildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda **"Hayır"** şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız **"Evet"** ise "Ölçme ve Değerlendirme"ye geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz..

1. Mikro işlemci ile mikro denetleyici arasındaki en ayırt edici özellik aşağıdakilerden hangisidir?
   A) Hafızaya erişim ana kart üzerindedir.
   B) Hafıza ile CPU aynı çekirdektedir.
   C) I/O, hafıza ve CPU aynı çekirdek içindedir
   D) Bütün birimler aynı ana karttadır.

2. Aşağıdaki modellerden hangisi mikro denetleyici değildir?
   A) 8051.
   B) ARM.
   C) MSP430
   D) 8088.

3. Mikro denetleyici içerisinde genel olarak aşağıdakilerden hangisi bulunmaz?
   A) ADC modülü
   B) VGA monitör çıkışı
   C) GPIO birimi
   D) PWM modülü.

4. Mikro denetleyicilerin işlem yapabilmesi için gerekli olan salınım (osilasyon) genel olarak aşağıdakilerden hangisi gerçekleştirir?
   A) Dâlilî veya haricî kristal
   B) +5V güç kaynağı
   C) ADC modülü
   D) PWM modülü.

5. Windows aygıt yöneticisinin görevini, genel olarak aşağıdakilerden hangisi anlatır?
   A) Dosyalama işlemlerini gerçekleştirir.
   B) Grafik ve ses dosyalarını yönetir.
   C) Sabit disk işlemlerini yönetir.
   D) Bilgisayarın donanım bilgilerini listeler.

6. Aşağıdakilerden hangisi mikro denetleyicileri programlayan (download) derleyici (compiler) olabilir?
   A) Delphi.
   B) IAR Systems
   C) C++
   D) Visual Basic.

7. Aşağıdakilerden hangisi seri iletişim için gerekli olan unsurlardan değildir?
   A) BaudRate.
   B) PortNo
   C) RJ45
   D) DataBits.

8. Seri iletişimde veri gönderme pini (uç – iğne) aşağıdakilerden hangisidir?
   A) TX.
   B) GND
   C) RX
   D) +5V

9. Aşağıdakilerden hangisi görsel programlama dili değildir?
   A) Delphi
   B) Visual Basic
   C) C++
   D) IAR Systems

10. Aşağıdakilerden hangisi görsel bileşen olan BUTTON bileşeninin olaylarındandır (event) ?
   A) KeyDown
   B) Click
   C) MouseMove
   D) Tick

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ–2

AMAÇ

Text dosyasındaki bilgiyi, betik dili ile okuyup, gerekli kod analizleri yaptırıp, sonuç bilgisini web'e aktarma işlemlerini gerçekleştirebileceksiniz.

ARAŞTIRMA

Text dosyalarından bilgi okuma, yazma, bulma işlemlerini web ortamında yapabilme kapasitesi olan bir web sitesini tasarlayınız.

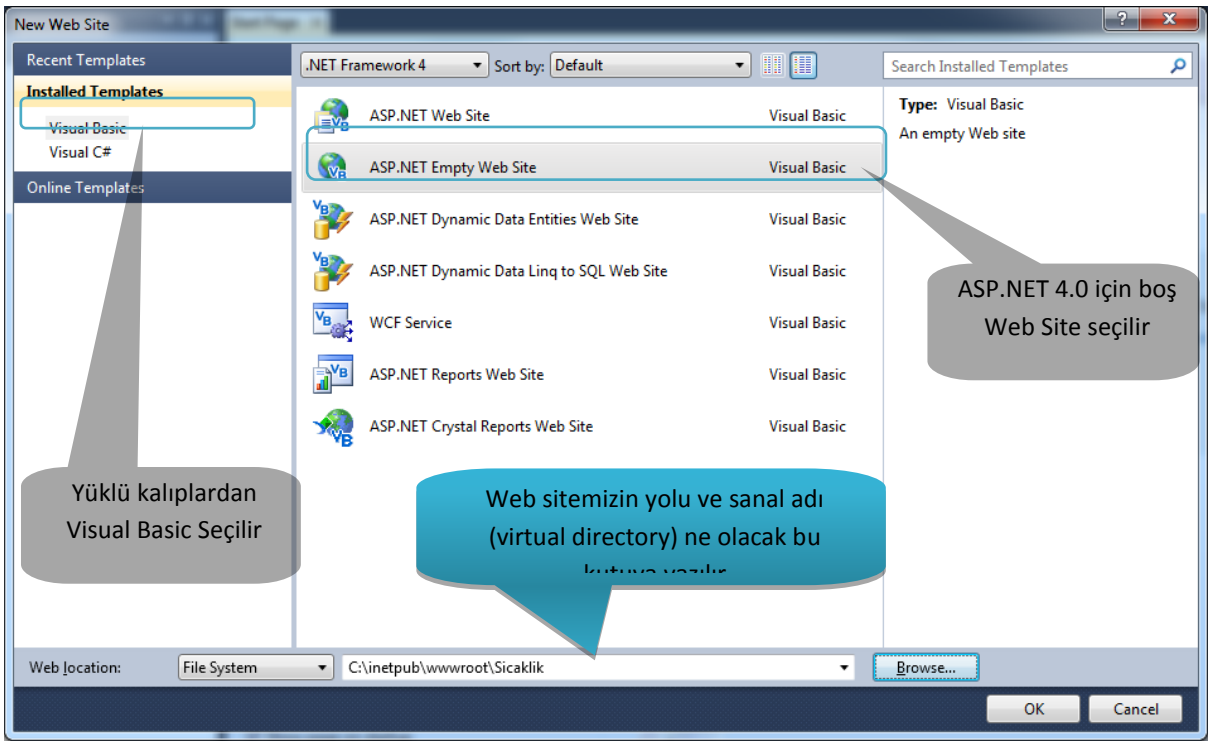
2. BETİK DİLİ İLE VERİLERİN WEB'E AKTARILMASI

Günümüzde web site tasarımlarında betik (script) dillerinde asp, asp.net, php veya programlama dillerinden Visual Basic, C#, DelphiPrism, RadPHP XE2 gibi çeşitleri bulunan yazılım araçlarıyla web sayfaları geliştirilmektedir. Özellikle programlama dil destekli web sayfa tasarımları programcılara büyük esneklik sağlamaktadır. Veri tabanı, dosya işlemleri ve seri veya USB port destekli donanıma yönelik kontrolleri kolay tekniklerle çözümünü getirmişlerdir.

2.1. Betik Dili Text Tipi Dosyadan Veri Okuma

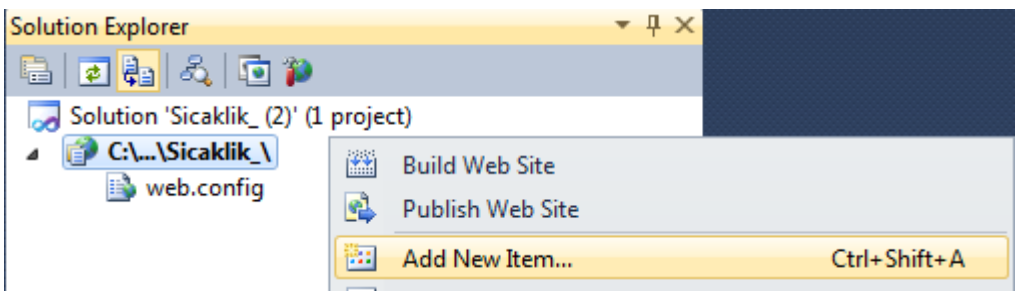
Uygulama projemiz olan Sıcaklık Okuma Otomasyon sistemi ile sonuc.txt dosyasına, kontrol kartında bulunan sıcaklık sensöründen elde ettiğimiz değeri İnsan Arayüz aracılığı ile okuyup, web sitemizi yayınladığımız sürücü ve yol olan **C:\inetpub\wwwroot\sicaklık** klasörüne kaydetme aşamalarını geçmiştik. Şimdi VS2010 VisualBasic yazılım aracılığı ile boş bir web projesi oluşturalım.

Resim 2.1 : Boş bir web site oluşturma ekranı (VS2010 Visual Basic)

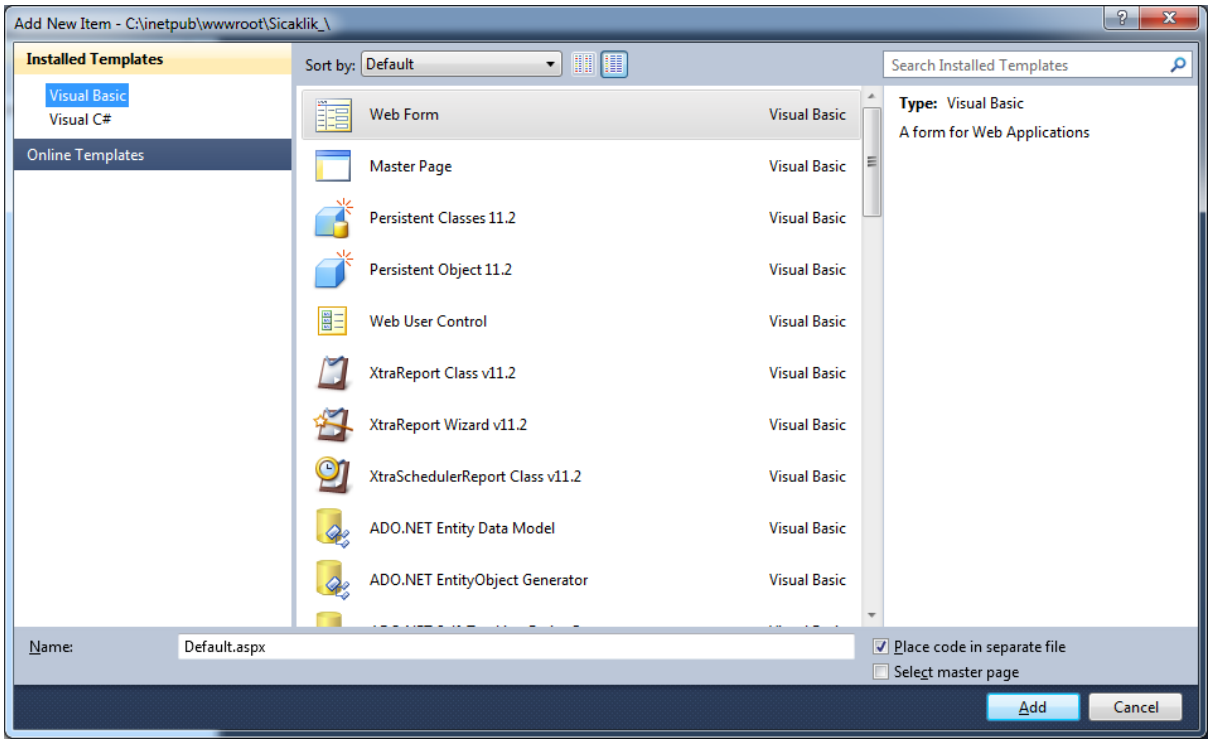


Resim 2.2 : Boş bir web sitemizin programlama dili, framework versiyonu (ASP.NET 4.0) ve sanal klasör bilgi girişleri ekranı

Oluşturulmuş boş web sitemize boş bir web sayfası eklemek için Solution Explorer penceresinden sağ fare tuşunu tıklamak suretiyle gelen menü penceresinden Add New Item seçeneği ile web form seçilir.



Resim 2.3 : Boş bir web sayfası oluşturma penceresei



Resim 2.4 : Web Form ve ismi belirtilip Add butonu ile boş web sayfası web sitemize eklenir

Web sitemize eklenen boş web sayfamıza (form) önceden elde ettiğiniz web tasarım bilgilerinizi kullanarak estetik tasarımlarla destekli bir veri okuma sayfası üretebilirsiniz. ASP.NET ile gelen Timer bileşeni ile otomatik veri okuma işlemlerini gerçekleştirebilirsiniz. Otomatik okuma işlemlerinde en önemli olay, **Timer** bileşeni ve okunan bilginin gösterildiği **Label** bileşeni. Bilgi çekmek için yazacağımız kodların olayı ise elbette **Timer** bileşenine ait **Tick** olayıdır. Örnek bir tasarımı altta inceleyebilirsiniz.

Resim 2.5 : Timer bileşen destekli sonuç.txt dosyasından veri okumak için örnek tasarım

Tasarımı table, resim ve renklerle daha estetik bir hale getirebilirsiniz. Bilgi çekme sıklığını ise **DropDownList** (**ComboBox** – Açılır kutu) içerisine **Items** olarak eklediğimiz 2-3-4 sn lik bilgi çekme sıklığını ifade eden bilgileri el ile (manuel) ekledik. Bu demektir ki **Timer** bileşeninin tetikleme aralığını her 2 sn de bir çalıştır demek istedik. Kullanıcı açılır kutu vasıtası ile 3 veya 4 sn lik zamanı da seçebilir. **Label7** bileşeninde ise

okunan bilgiyi görsel hâle dönüştürmek için kullanıldı.

Açılır kutu (DropDownList) bileşeninin bilgi değiştirildiği an tetiklenen (`Selected Index Changed`) olayına Timer bileşeninin tetikleme süresini ayarlayan Interval özelliğini kod ile değiştirme işlemi yapıldı.

```
Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles DropDownList1.SelectedIndexChanged

    Timer1.Interval = DropDownList1.SelectedItem.Value * 1000

End Sub
```

Resim 2.6 : Kullanıcı Açılır Kutu (DropDownList) bileşenindeki bilgiyi değiştirince çalışacak olan kodlar

Timer bileşeninin **Tick** olayındaki **algoritma** ise oldukça basittir. Tasarım sayfamızda da belirttiğimiz gibi, şimdiki zaman nedir, sıcaklık bilgisinin kayıt zamanı ve sıcaklık bilgisi bilgilerini analiz edip, gerekli görsel bileşenlere değerlerini atama işleminden ibarettir.

```
Protected Sub Timer1_Tick(ByVal sender As Object, ByVal e As System.EventArgs) Handles Timer1.Tick

    Label3.Text = System.DateTime.Now.ToString

    '************  TEXT DOSYASINDAN OKUMA İŞLEMİ BURADAN BAŞLIYOR *********************************
    Dim GelenBilgi As String = ""
    Dim DosyaAkisi As FileStream = New FileStream(Server.MapPath("sonuc.txt"), FileMode.Open)
    Dim AkisOku As StreamReader = New StreamReader(DosyaAkisi)
    GelenBilgi = AkisOku.ReadToEnd()
    Label5.Text = Mid(GelenBilgi, 1, 19)
    Label7.Text = Mid(GelenBilgi, 21, 4) + " °C"
    DosyaAkisi.Close()
    '*****************  TEXT DOSYASINDAN OKUMA İŞLEMİ BURADA BİTTİ  ****************************

End Sub
```

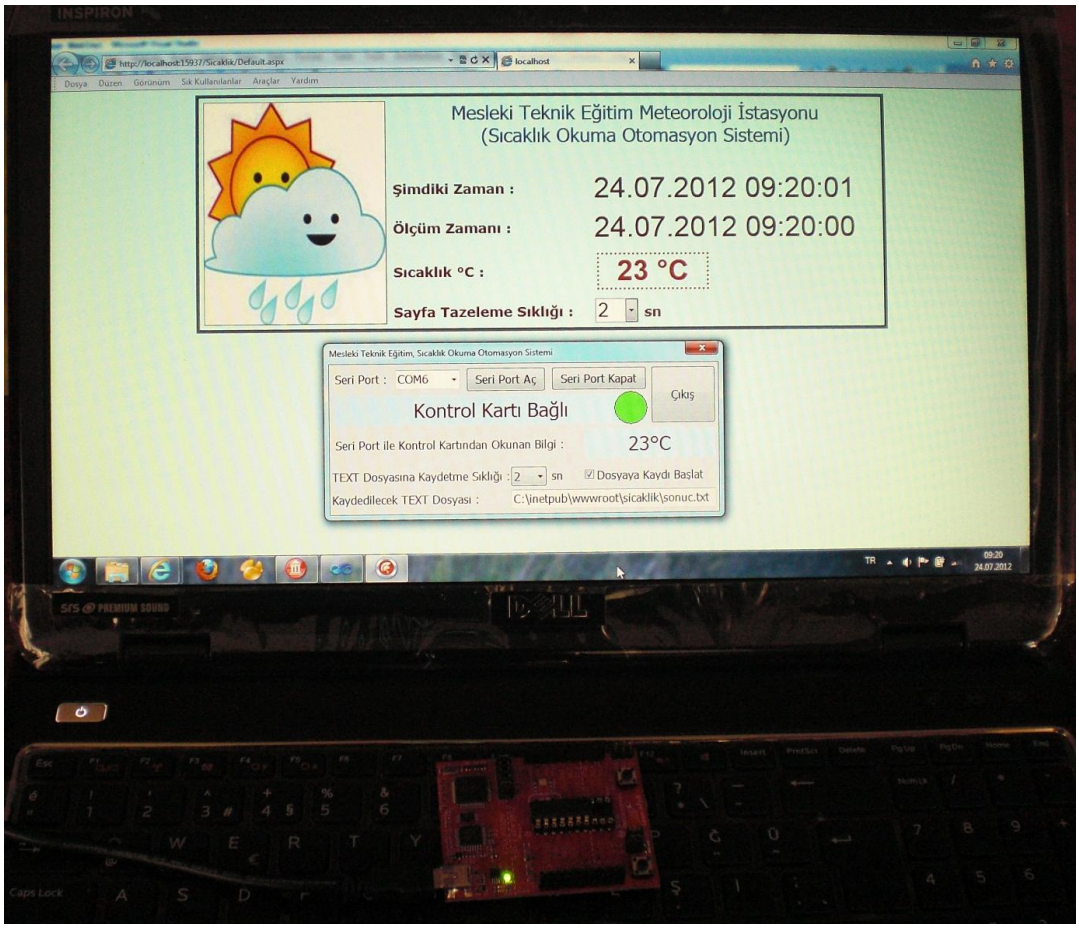
Resim 2.7 : Timer bileşenine ait Tick olay kodları

Sırasıyla değerlendirme yapalım;

- **Label3** bileşeninde sistem zamanını göster.
- **GelenBilgi** (String) değişkeni dosyadan okunan bilgiyi bu değişkene atayacağız.
- **DosyaAkisi** (FileStream) sanal klasörümüzde (web sitemizi yayınladığımız klasör) bulunan sonuc.txt dosyasını AÇ (**open**) modunda açarak kullanmamızı sağlar.

- AkisOku (StreamReader) DosyaAkisi değişkenindeki bilgileri okuyup bu değişkene bilgileri akıtır (atama işlemi). Bilgi RAM'da demektir.
- **GelenBilgi** değişkenine, **AkisOku** daki bilgileri dosya sonu oluncaya kadar okuyup ata.
- `Label5.Text = Mid(GelenBilgi, 1, 19)` Label5 te ilk 19 karakterlik bilgiyi göster. Bu bilgi görsel program tarafından sonuc.txt dosyasına kaydettiğimiz `07.07.2012 12:11:05 30` bilgisinden elde edilir. Yani kontrol kartında bulunan sıcaklık sensöründen elde edilen bilginin dosyaya kayıt zamanıdır. Şimdiki zaman değil.
- `Label7.Text = Mid(GelenBilgi, 21, 4) + " °C"` Label7 bileşeninde ise asıl istenen sıcaklık bilgisini web ortamında görmemizi sağlayan analiz bulunmaktadır. Sonuc.txt dosyasına kaydedilen ilk 20 karakterden sonraki bilgilerin tamamı sıcalık bilgisidir. Bu bilgiye ilaveten `°C` karakterleri de eklenerek daha bilimsel görmemizi sağlayacaktır.

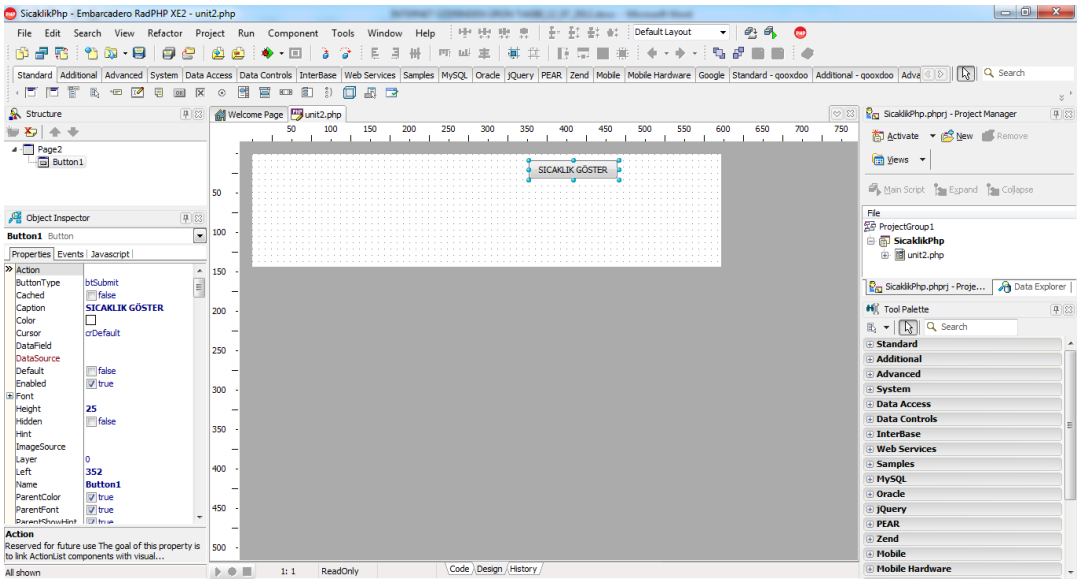
Web ortamında çalıştırdığımızda (Internet Explorer veya Mozilla) sadece sonuc.txt dosyasındaki bilgileri analiz sonuçlarına göre gösterir. Eğer, aynı anda görsel program ile kontrol kartınıda aynı bilgisayar üzerinde çalıştırdığınızda (Web Server da) interaktif olarak sıcaklık değişikliği; sensöre, sensörden usb port ile pc ye, görsel program ile sonuc.txt dosyasına oradan da web sitemize yol almış olur.



Resim 2.8 : Web server a bağlı mikro denetleyici, görsel program ve internet ortamında çalıştırılan sonuç.txt dosyasının durumu

2.2. Betik Dili İle Verilerin Web e Aktarılması

ASP.NET ile yapılan işlemlerin bir benzerini daha az görsellik kullanarak RadPHP XE2 yazılım aracıyla oluşturalım. Her ne kadar RadPHP editörü kullanılsa da php kodları geleneksel olduğu için herhangi bir php editörde aynı işlemler gerçekleştirilir.



Resim 2.9 : RadPHP XE2 oluşturulmuş ekran tasarımımız

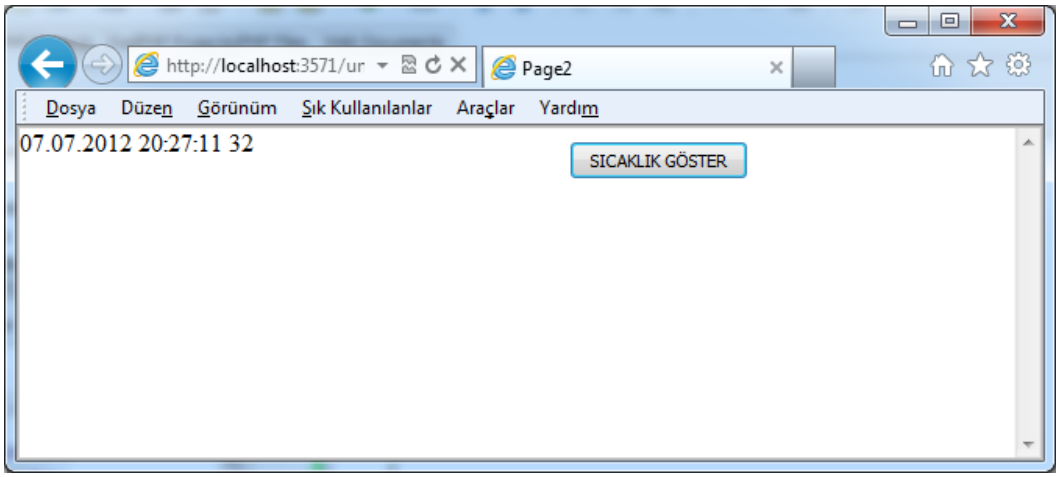
Ekranımızda sadece bir buton ve butona ait tıklama (Click) olayı vardır. Kullanıcı bu butonu tıkladığında aşağıdaki kod çalışır.

```
function Button1Click($sender, $params)
{
    $fp = fopen("C:\inetpub\wwwroot\Sicaklik\sonuc.txt", 'r');
    while(!feof($fp))
    {
      $veri = fgets($fp,999);
      echo $veri."<br>";
    }
  }
```

Resim 2.10 : Button bileşenin tıklama (click) olayı (RadPHP XE2)

```php
$fp = fopen("C:\inetpub\wwwroot\Sicaklik\sonuc.txt", 'r');
while(!feof($fp))
{
  $veri = fgets($fp,999);
  echo $veri."<br>";
}
```

Resim 2.11 : Geleneksel php kodları. Php destekli web sitenizde herhangi bir yerde kullanılan kodlar



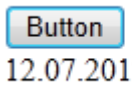
Resim 2.12 : Kullanıcı tarafından tıklanan butonun, php ile sonuç.txt dosyasından bilgi okuma işlemi

UYGULAMA FAALİYETİ

ASP.NET ile Tarih ve Saat Elde Etme Uygulaması

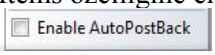
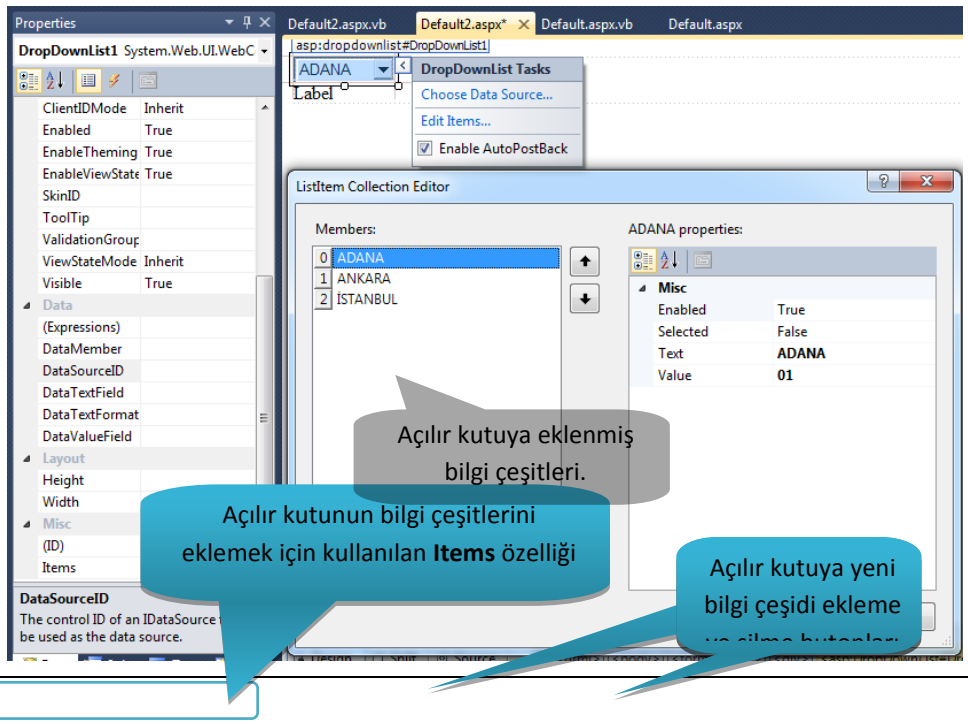
İşlem Basamakları
1. Boş bir ASP.NET 4.0 Visual Basic web projesi oluşturunuz.
2. Web formuna (web sayfası) Button ve Label getiriniz.

3. Button ın tıklama (click) olayına aşağıdaki kodları yazınız.
`Protected Sub Button1_Click(sender As Object, e As System.EventArgs) Handles Button1`
`    Label1.Text = System.DateTime.Now.ToString`
`End Sub`

4.  Butonu tıkladığımızda benzer bir ekran görmelisiniz.

UYGULAMA FAALİYETİ

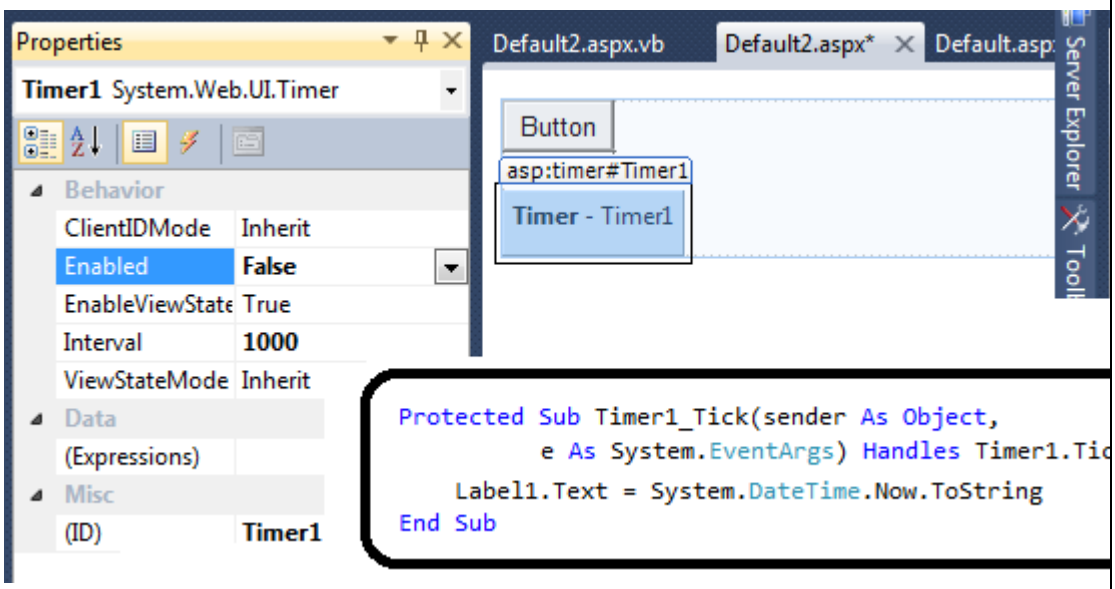
ASP.NET ile açılır kutuya (dropdownlist – combobox) manuel bilgi ekleyip, eklenmiş bilgi seçilince etiket (label) bileşeninde gösteren uygulama.

İşlem Basamakları
1. Boş bir ASP.NET 4.0 Visual Basic web projesi oluşturunuz.
2. Web formuna (web sayfası) Açılır kutu (DropDownList) ve Label getiriniz.
3. Açılır kutu (DropDownList) bileşenini seçip, Items özelliğine el ile (manuel) ADANA, ANKARA, İSTANBUL bilgilerini ekleyiniz.  kutusunu işaretlemeyi unutmayınız. Kullanıcı her seçim yaptığında web sayfası otomatik tazeleme işlemini gerçekleştirmek için kullanılır.
4. Açılır kutunun seçilen bilgiyi göster (`DropDownList1_SelectedIndexChanged`) olayına aşağıdaki kodlar yazılır.


UYGULAMA FAALİYETİ

ASP.NET ile etikette (Label) saati gösterme işlemini otomatik yapan web tasarımını gerçekleştiriniz (**Timer** kullanarak).

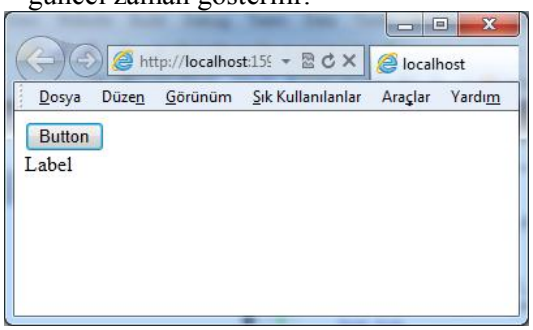
İşlem Basamakları
1. Boş bir ASP.NET 4.0 Visual Basic web projesi oluşturunuz.
2. Web formuna (web sayfası) Button, Timer ve Label getiriniz.
3. **Timer** bileşeninin özelliklerini (**properties**) aşağıdaki gibi ayarlayınız. Timer ilk başta pasif olmalı (Timer **Enabled = False**). Çalışma sıklığı (**Interval**) 1000 ms ye ayarlanmalı ve **Tick** (tetikleme) olayına ise aşağıdaki kodlar yazılmalı.

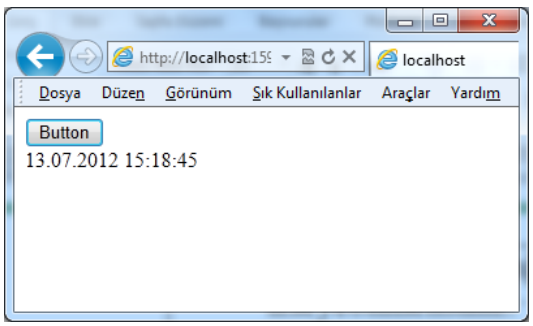


Butonun tıklama(click) olayına Timer bileşeninin Enable özelliğini kontrol edip aktif veya pasif yapmak suretiyle etiketteki (label) saat bilgisinin güncelleme durumunu kontrol altına almış oluruz.

```
Protected Sub Button1_Click(sender As Object, e _
    As System.EventArgs) Handles Button1.Click

    If Timer1.Enabled Then
        Timer1.Enabled = False
    Else
        Timer1.Enabled = True
    End If

End Sub
```

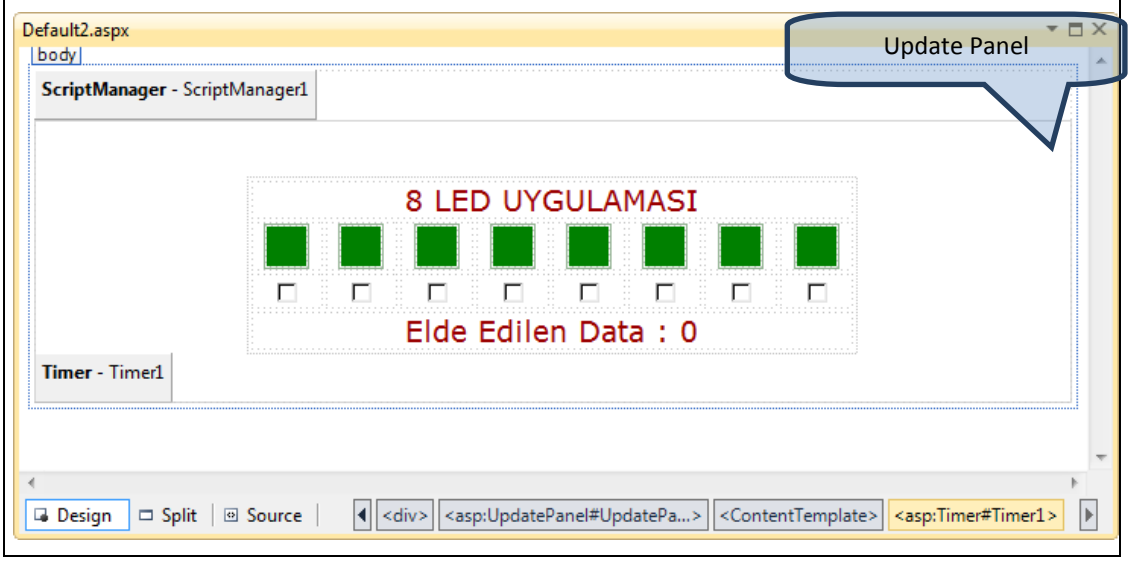
4. Sonuç olarak aşağıdaki ekran görüntülerinde verilen değerlere benzer bilgileri izleyebilmelisiniz. İlk çalışma anında etikette (label) bilgi gözükmez. Buton tıklandığında **Timer** aktif olur ve **Tick** olayına yazılmış kodlar işletilmeye başlanır. Etikette (label) güncel zaman gösterilir.





UYGULAMA FAALİYETİ

Web Programlama dili ile 8 bit veri elde etmek. Kullanıcı işaret kutularını (CheckBox) tıkladığından arka planda kodlar otomatik olarak 8 bit lik hesap sonuçlarını etikette (label) gösterme işlemidir.

İşlem Basamakları
1. Boş bir ASP.NET 4.0 Visual Basic web projesi oluşturunuz.
2. Web formuna (web sayfası) 1 Table (3 satır, 8 sütun), 1 Timer, 8 CheckBox (İşaret Kutusu), 8 panel (led için kullanılacak) ve 1 etiket (Label) getiriniz.
3. Web formuna aşağıdaki ekran görüntüsündeki tasarımı gerçekleştiriniz. Web formundaki (sayfasındaki) bileşenlerin özelliklerine kısaca göz atalım; Timer bileşeninin özellikleri (properties – Interval = 1000, Enable = True). Panel bileşeninin özellikleri (properties – BackColor = Green, Width = 30, Height = 30). İşaret kutusunun özellikleri (properties – Text = "boş- yazı yok"). AJAX destekli ScriptManager (ASP.NET için) ve Update Panel bileşenleri (otomatik tazelemeler için). Boş web formumuza (sayfamıza) bileşenlerimizi aşağıdaki gibi yerleştiriyoruz.   

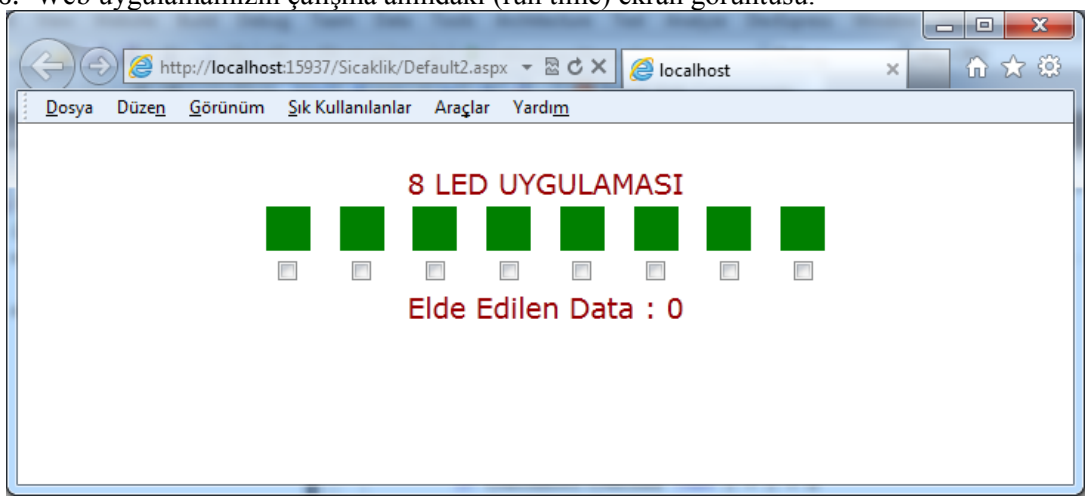
4. Timer bileşeninin Tick (otomatik tetikleme) olayına aşağıdaki kodları yazınız. **Timer** bileşenini **Tick** olayında; bütün panellerin (**LED**'ler) ilk geçerli (default) rengi yeşil (**Green**) yani led sönük olarak ayarlanır. Kullanıcı herhangi bir veya daha fazla işaret kutusunu (CheckBox) işaretlemiş ise işaretli olan kutunun hemen üstündeki panel rengini Açık Yeşil (**Lime**) olarak ayarlama işlemlerini gerçekleştirir ve **URETILEN_DATA** isimli prosedürü (**procedure** – **method**) tetikleyerek çalıştırır.

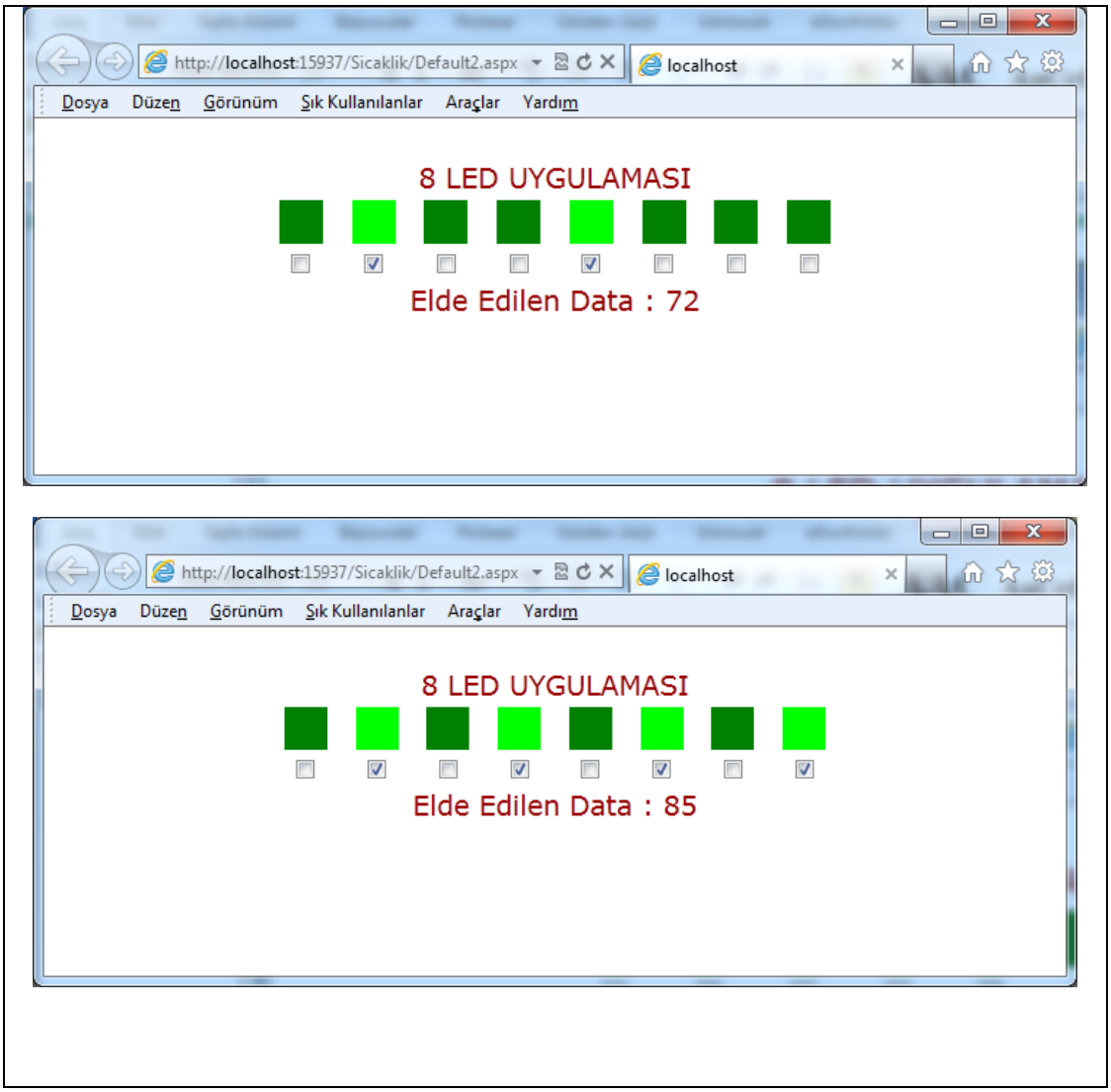
```
Protected Sub Timer1_Tick(sender As Object, e As System.EventArgs) Handles Timer1.Tick

    Panel1.BackColor = Drawing.Color.Green
    Panel2.BackColor = Drawing.Color.Green
    Panel3.BackColor = Drawing.Color.Green
    Panel4.BackColor = Drawing.Color.Green
    Panel5.BackColor = Drawing.Color.Green
    Panel6.BackColor = Drawing.Color.Green
    Panel7.BackColor = Drawing.Color.Green
    Panel8.BackColor = Drawing.Color.Green

    If CheckBox1.Checked Then Panel1.BackColor = Drawing.Color.Lime
    If CheckBox2.Checked Then Panel2.BackColor = Drawing.Color.Lime
    If CheckBox3.Checked Then Panel3.BackColor = Drawing.Color.Lime
    If CheckBox4.Checked Then Panel4.BackColor = Drawing.Color.Lime
    If CheckBox5.Checked Then Panel5.BackColor = Drawing.Color.Lime
    If CheckBox6.Checked Then Panel6.BackColor = Drawing.Color.Lime
    If CheckBox7.Checked Then Panel7.BackColor = Drawing.Color.Lime
    If CheckBox8.Checked Then Panel8.BackColor = Drawing.Color.Lime

    URETILEN_DATA()

End Sub
```

5. **URETILEN_DATA** isimli prosüdürümüzde (**procedure** – **method**) ise; I isimli **Integer** tipinde bir değişken tanımlanır. Her bir işaret kutumuzda (**CheckBox**) olan sabit değerlerimizi, eğer işaretli ise I değişkeninde biriktiren (toplayan) işlemleri gerçekleştirir.

```
Protected Sub URETILEN_DATA()

    Dim I As Integer = 0

    If CheckBox1.Checked Then I = I + 128
    If CheckBox2.Checked Then I = I + 64
    If CheckBox3.Checked Then I = I + 32
    If CheckBox4.Checked Then I = I + 16
    If CheckBox5.Checked Then I = I + 8
    If CheckBox6.Checked Then I = I + 4
    If CheckBox7.Checked Then I = I + 2
    If CheckBox8.Checked Then I = I + 1

    Label1.Text = "Elde Edilen Data : " + I.ToString

End Sub
```

6. Web uygulamamızın çalışma anındaki (run time) ekran görüntüsü.




http://localhost:15937/Sicaklik/Default2.aspx
localhost
Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım
8 LED UYGULAMASI
Elde Edilen Data : 72
http://localhost:15937/Sicaklik/Default2.aspx
localhost
Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım
8 LED UYGULAMASI
Elde Edilen Data : 85

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri **Evet**, kazanamadığınız becerileri **Hayır** kutucuğuna (**X**) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Boş bir web site ve sayfasını üretip, çalıştırabildiniz mi?		
2. Kodları yazdınız mı?		
3. Web sitesini web server da test ettiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda **"Hayır"** şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız **"Evet"** ise "Ölçme ve Değerlendirme"ye geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi betik dillerinden değildir?
   A) ASP
   B) Java
   C) C++
   D) C#

2. Windows web server kurulu bilgisayarda ana (kök) sürücü ve klasör aşağıdakilerden hangisidir?
   A) wwwroot
   B) inetpub
   C) inetpub\wwwroot
   D) C:\inetpub\wwwroot

3. ASP.NET FrameWork 4.0 ile üretilmiş web sitelerindeki, **web formlarının** (web sayfaları) uzantısı aşağıdakilerden hangisidir?
   A) Aspx
   B) asp
   C) php
   D) c#

4. Hem görsel programlamada hem de web programlamada kullanılan **Timer** (zamanlayıcı) bileşeninin **en belirleyici** özelliği aşağıdakilerden hangisidir?
   A) Kullanıcı işlemi gerektirmez.
   B) Otomatik olay tetikleyicisi vardır.
   C) Aynı anda bir çok işlem yapabilir.
   D) Tıklama (click) olayı yoktur.

5. Web programlama ile oluşturduğumuz projemizi (web sitemizi) hangi arayüz programı **çalıştırmaz**?
   A) Safari
   B) Mozilla
   C) Excel.
   D) Internet Explorer

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise "Modül Değerlendirme"ye geçiniz

MODÜL DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Mikro denetleyiciler çok gelişmiş bilgisayarlarda kullanılan işlemcilerdir.
2. ( ) Mikro denetleyiciler dış dünyadan analog bilgi toplar.
3. ( ) Mikro denetleyiciler üzerinde sıcaklık sensörleri bulunabilir.
4. ( ) Mikro denetleyicilerde mantıksal (Logic) 0 demek +3V veya +5V demektir.
5. ( ) Mikro denetleyicileri programlamak için Visual Basic yazılım aracı kullanılır.
6. ( ) 115200 Baud paralel portlarda bilgi iletişim hız birimidir.
7. ( ) C programlama dilinde sonsuz döngü kodu : ***while (1)***'dir.
8. ( ) C programlama dilinin en temel kalıp kodu : ***void main(void) { while (1) }***'dir
9. ( ) Web programlamada kullanılan betik dili ***ASP.NET*** dilidir.
10. ( ) ***php, asp,*** betik dillerine örnek web teknolojileridir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	**C**
2	**D**
3	**B**
4	**A**
5	**D**
6	**B**
7	**C**
8	**A**
9	**D**
10	**B**

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	**C**
2	**D**
3	**A**
4	**B**
5	**C**

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

1	**Y**
2	**D**
3	**D**
4	**Y**
5	**Y**
6	**Y**
7	**D**
8	**D**
9	**Y**
10	**D**

KAYNAKÇA

- **Eğitim Bilişim Ağı**, e-İçerik, Çizgi Tagem, www.eba.gov.tr (18.07.2012).
- **IAR SYSTEMS**, gömülü sistemler için yazılım araçları üreticisi, www.iar.com (18.07.2012).
- **MSP 430 Data Sheet Texas Instruments**, mikro işlemci üreticisi, www.ti.com (18.07.2012).
- **MCU TURKEY**, Gömülü Sistem Projeleri ve Destek Programı, www.mcu-turkey.com (18.07.2012)
- **Embarcadero**, C++, Delphi, RadPHP programlama dili üreticisi www.embarcadero.com (18.07.2012).
- **Microsoft** C#, Visual Basic, asp, ASP.NET programlama dilleri ve web teknolojileri üreticisi www.microsoft.com (18.07.2012).